

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Saša Špiler

**Pregled ogrodij za hiter razvoj
podatkovno usmerjenih aplikacij**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Dejan Lavbič

Ljubljana 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Na svetovnem spletu obstajajo številni med seboj nepovezani podatkovni viri, ki bi z inteligentnim pristopom povezavanja lahko prinesli veliko dodano vrednost v poslovnih okoljih, kot tudi končnim uporabnikom. Poslovnim in tehničnim uporabnikom so na voljo številni pristopi in ogrodja, ki omogočajo hitrejši razvoj takšnih integracij podatkov in vizualizacijo rezultatov. V okviru diplomske naloge preglejte različna ogrodja, ki omogočajo hiter razvoj podatkovno usmerjenih aplikacij in izdelavo t.i. mash-up aplikacij. Osredotočite se na ključne predstavnike in jih kritično ovrednotite glede na kriterije, ki pomembno vplivajo na razvoj takšnih aplikacij. V okviru evalvacije implementirajte izbrano problemsko domeno v ogrodjih, ki se izkažejo za najboljša.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana Saša Špiler, z vpisno številko **63100424**, sem avtor diplomskega dela z naslovom:

Pregled ogrodiš za hiter razvoj podatkovno usmerjenih aplikacij

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom doc. dr. Dejana Lavbiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 13. septembra 2014

Podpis avtorja:

Iskreno bi se zahvalila svojemu mentorju doc. dr. Dejanu Lavbiču za posvečen čas, svetovanje in pomoč pri izdelavi diplomskega dela. Zahvaljujem se tudi Špeli Pokovec za lektoriranje.

Vsem, ki verjamete vame.

Kazalo

Seznam uporabljenih kratic

Povzetek

Abstract

1	Uvod	1
2	Prepletene storitve	3
2.1	Arhitektura	4
2.2	Prednosti in slabosti prepletenih storitev	5
3	Kategorizacija prepletenih storitev	7
3.1	Tehnološki nivo prepletenih storitev	8
3.2	Okolje izvajanja	11
3.3	Namen prepletenih storitev	15
3.4	Ciljni uporabniki	16
3.5	Podpora uporabnikom in soustvarjanje vsebine	17
4	Pregled ključnih ogrodiš za razvoj podatkovno usmerjenih aplikacij	21
4.1	Apatar	21
4.2	Dapper	23
4.3	Google Spreadsheets	24

4.4	WireCloud	25
4.5	Yahoo Pipes	27
5	Večkriterijski odločitveni model za izbiro ogrodja za razvoj podatkovno usmerjenih aplikacij	29
5.1	Opis kriterijev	30
5.2	Rezultati	34
6	Implementacija prepletene storitve	39
6.1	Prepletena storitev v ogrodju Yahoo Pipes	39
6.2	Prepletena storitev v ogrodju Google Spreadsheets	45
6.3	Primerjava implementacije	48
7	Sklepne ugotovitve	51
	Literatura	52

Seznam uporabljenih kratic

API (Application Programming Interface) je aplikacijski programski vmesnik.

AJAX (Asynchronous JavaScript and XML) je skupina medsebojno povezanih spletnih razvojnih tehnik, uporabljenih za ustvarjanje interaktivnih spletnih aplikacij.

CSS (Cascading Style Sheets) so stilne predloge, predstavljene v obliki preprostega slogovnega jezika, ki skrbijo za prezentacijo spletnih strani.

DaaS (Device as a Service) je naprava kot storitev.

DIY (Do It Yourself) je metoda naredi sam, ki zajema razvoj, spreminjanje ali popravilo brez nadzora specialistov.

ETL (Extract, Transform, Load) je proces, ki zbere podatke iz zunanjega vira, jih prilagodi operativnim potrebam in jih naloži na ciljni naslov.

HTML (Hyper Text Markup Language) je označevalni jezik za izdelavo spletnih strani.

HTTP (HyperText Transfer Protocol) je protokol za izmenjavo hiperteksta ter grafičnih, zvočnih in drugih večpredstavnostnih vsebin na spletu.

JSON (JavaScript Object Notation) je format za izmenjavo podatkov, ki temelji na strukturi zbirke parov ključ/vrednost.

RSS (Rich Site Summary) je družina datotečnih oblik XML za spletno zlaganje, ki ga uporabljajo novičarske spletne strani ali spletni dnevniki.

REST (Representational State Transfer) je arhitekturni stil za izmenjavo podatkov med spletnimi storitvami.

SaaS (Software as a Service) je programska oprema kot storitev.

SQL (Structured Query Language) je strukturirani povpraševalni jezik za delo s podatkovnimi bazami.

SOAP (Simple Object Access Protocol) je protokol za izmenjavo strukturiranih podatkov in implementacijo spletnih storitev v računalniških omrežjih.

XHTML (eXtensible Hyper Text Markup Language) je označevalni jezik, ki ima enak namen kot HTML, vendar je skladen s sintakso XML.

XML (eXtended Markup Language) je razširljiv označevalni jezik.

Povzetek

Na svetovnem spletu obstajajo številni med seboj nepovezani podatkovni viri, odprti programski vmesniki in storitve, ki jih je mogoče s pomočjo različnih ogrodij povezovati v prepletene storitve. Njihov začetek sovпада s pojavom spletnih aplikacij druge generacije znanih kot Splet 2.0. V diplomskem delu so ogrodja za hiter razvoj podatkovno usmerjenih aplikacij kategorizirana glede na svoje ključne lastnosti in medsebojne razlike. Izbrani predstavniki so podrobneje raziskani ter nato ocenjeni znotraj večkriterijskega odločitvenega modela na podlagi uporabniške izkušnje, podpore uporabnikov in funkcionalnosti. V najboljše ocenjenih ogrodjih Yahoo Pipes in Google Spreadsheets je zgrajena prepletena storitev, ki uporabniku prikaže dodatne informacije o izbrani državi v obliki zemljevida, videoposnetkov ter novic.

Ključne besede: prepletene storitve, ogrodja, hiter razvoj, Yahoo Pipes, Google Spreadsheets, Splet 2.0.

Abstract

On the World Wide Web there are many unrelated data sources, application programming interfaces and services. Nowadays it is possible to link them with the help of different tools to a unified mashup application. Their appearance coincides with the start of the second generation of the web called Web 2.0. In this thesis tools for quick development of data-oriented applications are put in different categories according to their mutual properties and differences. Selected representatives are later on explored and assessed in a multi-attribute decision model according to their user experience, user support and functionalities. Based on results of the model a mashup is developed in the Yahoo Pipes and the Google Spreadsheets tool. With this mashup, the user is able to see additional information about the selected country including map, videos and news feed.

Keywords: mashups, tools, quick development, Yahoo Pipes, Google Spreadsheets, Web 2.0.

Poglavje 1

Uvod

V zadnjih letih uporaba interneta še vedno strmo narašča. S pojavom spleta druge generacije, imenovanega Splet 2.0, so uporabniki prevzeli novo vlogo, v kateri so postali še aktivnejši udeleženci znotraj njega, predvsem z možnostjo soustvarjanja vsebin. Nekatera podjetja so omogočila prosto uporabo svojih storitev in podatkovnih virov, ki so postali dostopnejši uporabnikom. Na tem mestu so se prvič pojavile tako imenovane prepletene storitve. Z njihovo pomočjo je uporabnik dobil možnost povezave podatkovnih virov in vmesnikov v enotno aplikacijo, katere razvoj ni zahteval programerskega znanja. Pripomogle so tudi k temu, da dobijo že obstoječi podatki novo uporabniško vrednost, saj lahko služijo reševanju in analiziranju večje problemske domene.

Skupaj s povečevanjem števila odprtih vmesnikov in podatkovnih virov je naraščalo tudi število ogrodij za njihovo povezovanje. Tako so nastala številna ogrodja, ki so specializirana in prilagojena različnim uporabniškim potrebam in znanju. Namenjena so tako osebnim kot tudi profesionalnim rabi. Večina ogrodij že vsebuje obstoječe gradnike za razvoj prepletenih storitev, kar olajša njihovo uporabo. Skupaj z možnostjo razvoja lastnih komponent pa so pridobila tudi bolj izkušeno skupino uporabnikov.

Dandanes skoraj vsak uporabnik spleta uporablja različne storitve, ki jih internet ponuja. Ogled posnetkov na YouTube, iskanje lokacij s pomočjo Google

Zemljevidov ali spremljanje slavnih na Twitterju. To, da je mogoče omenjene storitve na enostaven način združiti v eno samo aplikacijo, je bilo še nedolgo nazaj težko verjeti. Njihov obstoj je tudi meni vzbudil zanimanje in čeprav sem v preteklosti že nekajkrat tudi sama uporabila tovrstne aplikacije, nisem nikoli prej razmišljala o njihovem ozadju, razvoju in večnamenskosti. To je bil povod, da sem se odločila za podrobnejši razvoj opisane tematike.

Cilj diplomske naloge je podrobno raziskati obstoječa ogrodja za razvoj prepletenih storitev. Ugotoviti želimo, v kakšne skupine jih lahko uvrstimo glede na njihove lastnosti ter katere so njihove medsebojne podobnosti in razlike. Pomembno je tudi najti neke splošne kriterije, na osnovi katerih lahko ocenjujemo ogrodja. Pričakujemo, da nam bodo ti pomagali določiti smerne, ki služijo uporabnikom pri njihovi izbiri za razvoj prepletenih aplikacij. Ker je obstoječih ogrodij cela množica, je naš cilj izmed teh najti reprezentativne predstavnike, ki si zaslužijo podrobnejšo analizo. Samo kategoriziranje in raziskovanje posameznih ogrodij pa še vedno ni dovolj za njihovo popolno razumevanje in predstavo o njih. To je razlog, da želimo ob koncu našega dela tudi razviti prepleteno storitev v enem ali več ogrodjih.

Diplomsko delo je razdeljeno na sedem poglavij. V uvodnem poglavju sta predstavljena tema in namen. Drugo poglavje opisuje, kaj so prepletene storitve z namenom lažjega razumevanje celotnega dela. Sledi tretje poglavje, kjer je opisana delitev prepletenih storitev glede na različne kategorije. Četrto poglavje je namenjeno pregledu izbranih ogrodij z informacijami o njihovem izgledu in uporabi. Večkriterijski odločitveni model, narejen v petem poglavju, na osnovi odločitvenega drevesa ovrednoti v prejšnjem poglavju predstavljena ogrodja glede na tri glavne kriterije. Šesto poglavje je namenjeno praktičnemu delu – narejeni prepleteni storitvi. Sledi še zaključek, ki vsebuje končne ugotovitve diplomskega dela.

Poglavje 2

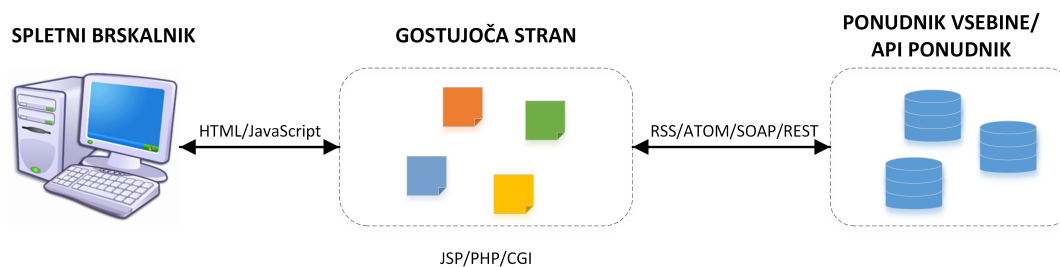
Prepletene storitve

Prepletena storitev (ang. *mashup*) je aplikacija, ki združuje podatke z uporabo programskih vmesnikov (ang. *API*) in drugih virov v eno orodje in uporabniško izkušnjo. Notranji podatkovni viri, viri RSS/Atom, spletno luščenje podatkov in spletne storitve so nekateri izmed virov, ki jih prepletene storitve lahko vključujejo [32, 34]. Primer prepletene storitve je aplikacija, ki na Googleovih zemljevidih prikaže najbolj znane kavarne po svetu skupaj z njihovimi fotografijami in mnenji obiskovalcev.

Aplikacijski programski vmesnik [21] je zbirka funkcij, procedur in razredov za dostopanje do spletne storitve ali vsebine, za razvoj spletnih storitev. Ločimo dve vrsti: (1) *lastniške* in (2) *odprtokodne*. Prve so tiste, katerih uporaba zahteva plačilo in pridobitev licence. Pri odprtokodnih gre običajno za brezplačne vmesnike, a imajo na račun tega omejujoče pogoje uporabe ali omejeno število dostopov od spletne storitve do ponudnika [9]. Nekateri izmed ponudnikov aplikacijskih programskih vmesnikov so Google, YouTube, Flickr, Twitter, Amazon in eBay.

2.1 Arhitektura

Prepletene storitve sestavlja trinivojska arhitektura, predstavljena na sliki 2.1. Ločimo naslednje tri nivoje [28]:



Slika 2.1: Tri-nivojska arhitektura prepletene storitve

1. Ponudnik vsebine ali API-ponudnik

To so ponudniki vsebine in podatkov, ki so integrirani znotraj prepletenih storitev. V namen poenostavitve procesa pridobivanja podatkov je ta vsebina na voljo prek protokolov, kot so REST, spletne storitve ali RSS/Atom.

2. Gostujoča stran prepletene storitve

Aplikacijska logika se lahko izvede na strani strežnika ali odjemalca. Na strani strežnika se za generiranje dinamične vsebine uporablja tehnologije, kot so Java servleti, CGI, PHP ali ASP. Alternativna možnost je, da se vsebina prepletene storitve generira neposredno znotraj uporabniškega brskalnika prek uporabniškega skriptiranja ali apletov.

3. Spletni brskalnik

Znotraj spletnega brskalnika poteka uporabniška interakcija, kjer postane storitev grafične narave.

2.2 Prednosti in slabosti prepletenih storitev

S hitrim povečevanjem prepletenih storitev, z nastajanjem novih ogrodij ter aplikacijskih programskih vmesnikov je treba raziskati tudi prednosti in slabosti, ki se pri tem pojavijo.

Prednosti prepletenih storitev [9]:

1. Ponovna uporaba

Ponudniki prepletenih storitev navadno ponujajo možnost ponovne uporabe storitev, ki so jih ustvarili drugi uporabniki. To je velika prednost, kajti uporabnik tako ni omejen na razvoj od čistega začetka, ampak lahko uporabi druge že razvite prepletene storitve in jih po potrebi tudi združuje.

2. Hiter razvoj

Nekatera ogrodja za razvoj prepletenih storitev imajo že na voljo komponente in operatorje za uporabo. Poleg tega programski vmesniki omogočajo dostop do zunanjih storitev. Vse to omogoča hitrejši razvoj prepletenih storitev.

3. Prilagojenost uporabniškim potrebam

Prepletene storitve ponujajo prilagojenost uporabniškim potrebam, saj jim omogočajo vključitev vsebine in že obstoječih vmesnikov znotraj njih. Uporabniku tako ni treba samemu razvijati omenjenih komponent.

4. Zmanjšanje stroškov

Kljub opisanim prednostim pri razvoju prepletenih storitev se je treba zavedati tudi negativnih posledic, ki lahko nastanejo. Na podlagi teh predpostavk se lahko uporabnik odloči, ali je problemska domena ustrezna za tovrstno realizacijo.

Slabosti prepletenih storitev [9]:

1. **Nezanesljivost ponudnikov vsebin in programskih vmesnikov**

Ogrodja za razvoj prepletenih storitev in ponudniki programskih vmesnikov ne zagotavljajo neomejeno trajajoče uporabe in podpore.

2. **Skalabilnost**

Navadno je uporabnik pri brezplačnih storitvah omejen na določeno število zahtev.

3. **Integriteta podatkov**

Za podatke ni nobenega zagotovila, da se bodo ohranili brez sprememb, poškodb ali celo izgub.

4. **Dostopnost samo spletnih podatkov**

V prepletene storitve je mogoče na enostaven način vključiti spletno dostopno programsko opremo. Večjo težavo za vključitev pa predstavljajo namizne aplikacije.

5. **Varnost**

Neobstoj standardov za prepletene storitve je razlog za oteženo implementacijo varnostnih mehanizmov. Problem varnosti predstavlja težavo tudi za podjetja z občutljivimi podatki, ki morajo preprečiti to, da bi integrirane vsebine lahko predstavljale grožnjo.

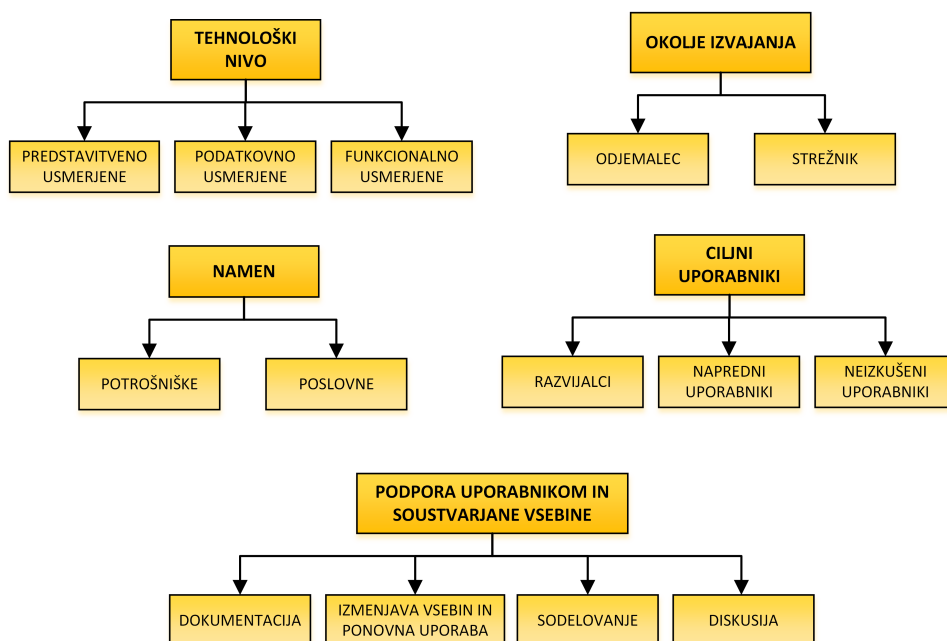
Poglavje 3

Kategorizacija prepletenih storitev

V tem poglavju so opisani splošni kriteriji za kategorizacijo ogrodič za ustvarjanje prepletenih storitev, ki jih prikazuje slika 3.1. V nadaljevanju je podrobneje opisan vsak izmed njih skupaj s podanimi tipičnimi predstavniki.

3.1 Tehnološki nivo prepletenih storitev

Tehnološki nivo pove, kakšne vrste storitev so integrirane znotraj posameznih ogrodič. Prepletene storitve so namreč ustvarjene z združevanjem in obdelavo storitev, pridobljenih iz različnih spletnih virov. Glede na to lahko ogrodič pripadajo naslednjim tehnološkim nivojem: (1) *predstavitvenemu*, (2) *podatkovnemu* ali (3) *funkcionalnemu*. Vsakemu izmed njih pripadajo tudi določeni programski jeziki in metode za ustvarjanje prepletenih storitev.



Slika 3.1: Kategorizacija glede na različne dimenzije

3.1.1 Predstavitveno usmerjene prepletene storitve

Predstavitveno usmerjene prepletene storitve lahko imenujemo tudi storitve za interakcijo z uporabnikom. Navadno so najenostavnejše in najhitreje za razvoj, saj zahtevajo malo ali skoraj nič programerskega znanja. Razvijalec enostavno izbira med že obstoječimi komponentami ter jih postavlja na grafični uporabniški vmesnik z uporabo načina »povleci in spusti« (ang. *drag and drop*).

Predstavitveno usmerjene prepletene storitve temeljijo na podatkih in komponentah uporabniškega vmesnika, ki so pridobljeni z uporabo storitvenih programskih vmesnikov z različnih spletnih strani. Tovrstni model imenujemo SaaS¹.

Tovrstne prepletene storitve navadno ne zahtevajo predhodne avtorizacije, namestitvenih korakov ali uporabe dodatnih tehnologij. Njihova implemen-

¹Programska oprema kot storitev.

tacija je enostavna zaradi uporabe javno dostopnih storitev, komponent in knjižnic, ki jih je mogoče prilagoditi lastnim potrebam. Zunanje spletne strani so tiste, ki skrbijo za postavitve programske rešitve za delovanje v produkcijskem okolju. Predstavitveno usmerjene prepletene storitve so tudi dobro odzivne, saj so vsi zahtevki naslovljeni neposredno na ponudnika storitev [18, 24].

Tehnologije ali orodja, ki jih lahko predstavitveno usmerjene prepletene storitve vključujejo [24]:

- **Pripomočki in gradniki**

Ti so lahko postavljeni na grafični uporabniški vmesnik neodvisno drug od drugega. Pojem gradniki zajema poleg običajnih elementov (npr. gumbi, drsniki, orodne vrstice) še elemente z dodatnimi funkcionalnostmi (npr. ura, kalkulator). Slednji pridobivajo informacije iz zunanjih podatkovnih virov.

- **JavaScript**

Na aplikacijsko ploščo lahko vstavimo del kode JavaScript, da ustvari komponente uporabniškega vmesnika.

- **CSS/HTML/XHTML**

S posameznimi deli kode CSS/HTML/XHTML lahko ustvarimo samostojne komponente uporabniškega vmesnika, ki jih lahko ponovno uporabimo neodvisno od problemske domene.

- **Komponente Flash / apleti Java / kontrole ActiveX**

Pri predstavitveno usmerjenih prepletenih storitvah je glavni poudarek predvsem na tem, kako so te predstavljene na spletu. Iz tega sledi, da smo se osredotočili predvsem na spletne tehnologije, omenjene v zgornjem seznamu.

ogrodje/nivo	predstavitveni	podatkovni	funkcionalni
Apatar [2]		x	
Dapper [5]	x		
DERI pipes [6]		x	
Google Spreadsheets [7]	x	x	
MarcoFlow [8]	x	x	x
Microsoft Excel [10]	x	x	
RSSBus [11]	x		
Yahoo Pipes [15]		x	
YQL [17]		x	
We Wired Web [12]	x	x	
Wirecloud [14]	x	x	

Tabela 3.1: Kategorizacija ogrodij glede na tehnološki nivo

3.1.2 Podatkovno usmerjene prepletene storitve

Podatkovno usmerjene prepletene storitve vključujejo dostopanje do podatkov iz različnih podatkovnih virov in njihovo integracijo v enem oknu ali strani, tipično z uporabo programskih jezikov (npr. JavaScript). Podatki so nato na voljo v obliki XML, RSS, JSON ali Atom.

Večina teh podatkovnih virov izvira iz podatkovno usmerjenih storitev, ki jih označujemo kot DaaS². Te opisujejo storitve programskega vmesnika, ki jih lahko kličemo brez pomoči zunanjih procesov med ponudnikom in stranko. Pri tem je treba omeniti še, da podatkovni viri lahko vključujejo strukturirane, delno strukturirane ali nestrukturirane podatke. Problemi, ki jih je treba upoštevati pri integraciji teh podatkov, so tako predvsem strukturne in semantične narave [18, 29, 24].

²Naprava kot storitev

3.1.3 Funkcionalno usmerjene prepletene storitve

Funkcionalno usmerjene prepletene storitve združujejo podatke in aplikacijske funkcionalnosti v enega ali več zunanjih procesov z uporabo programskih jezikov, kot so Java, PHP, Python ali C++. Funkcionalnosti so dostopne prek programskih vmesnikov in integrirane z uporabo medprocesnih komunikacijskih tehnik (npr. skupen spomin, sporočilne vrste, klic oddaljenih procedur – RPC). Od podatkovno usmerjenih prepletenih aplikacij se razlikuje v tem, da je končni rezultat tukaj nov proces ali storitev, medtem ko je pri podatkovno usmerjenih rezultat nov podatkovni model [24, 27].

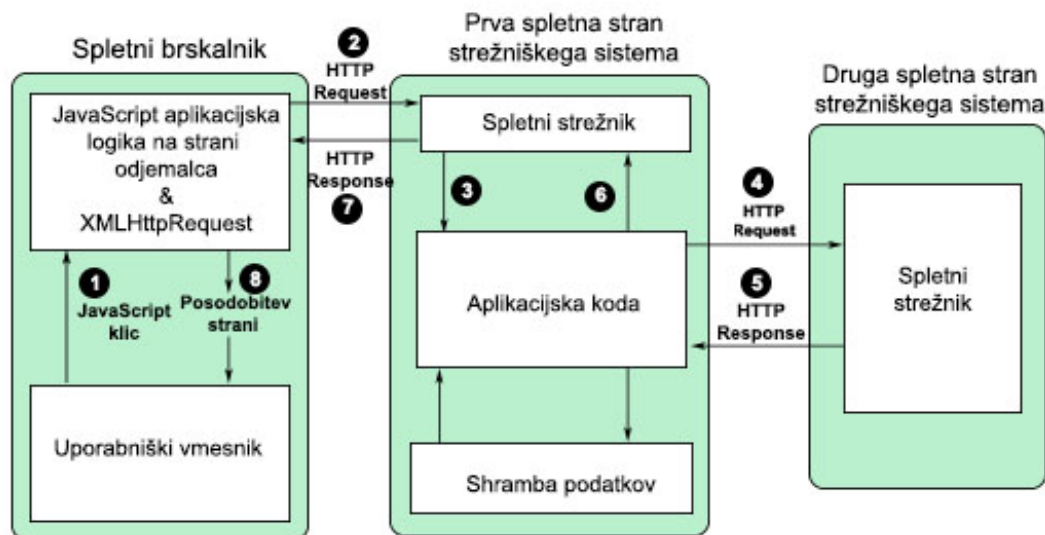
Obstaja veliko število ogrodij, ki pripadajo enemu ali več iz omenjenih tehnoloških nivojev. Na tem mestu je omenjenih le nekaj tipičnih predstavnikov, ki so prikazani v tabeli 3.1. Opazimo, da se najpogosteje pojavlja podatkovni nivo s šestimi pripadniki, najredkeje pa funkcionalni nivo.

3.2 Okolje izvajanja

Izvajalno okolje prepletenih storitev ločimo glede na to, ali se izvajajo na: (1) *strežniku* ali (2) *odjemalcu*. Navadno je za ustvarjanje prepletenih storitev uporabljena kombinacija obeh pristopov [27]. V nadaljevanju bosta podrobneje opisani obe izvajalni okolji.

3.2.1 Prepletene storitve na strani strežnika

Pri tem pristopu so storitve in vsebina integrirane na strežniku. Te se nato prenesejo do odjemalca prek protokola HTTP. Tako ima strežnik vlogo posrednika med spletno aplikacijo na strani odjemalca (brskalnikom) in drugo spletno stranjo. Vsi zahtevki gredo od odjemalca k strežniku, ki izreja klice do zelenih spletnih strani. [19, 27].



Slika 3.2: Prepletena storitev na strani strežnika [30]

Proces, ki opisuje delovanje prepletene storitve na strani strežnika (glej sliko 3.2):

1. *Uporabnik proži dogodek na odjemalcu, tipično brskalniku.*
2. *Odjemalec naredi zahtevek `Ajax XmlHttpRequest` do strežnika.*
3. *Program na strežniku prejme zahtevek in kliče metodo, ki vsebuje kodo za povezavo z drugimi spletnimi stranmi znotraj prepletene storitve.*
4. *Razred posredniškega strežnika obdela zahtevek in odpre stran, ki vsebuje zahtevano storitev.*
5. *Stran, ki vsebuje prepleteno storitev, prejme zahtevo v obliki `HTTP GET` ali `HTTP POST`, obdela zahtevek in vrne dobljene podatke razredu posredniškega strežnika.*

6. *Razred posredniškega strežnika prejme odgovor in ga pretvori v odjemalcu ustrezen podatkovni format.*
7. *Program na strežniku vrne odgovor odjemalcu.*
8. *Povratni klic vsebovan v XMLHttpRequest posodobi pogled strani za odjemalca z manipulacijo dokumentnega objektnega modela (DOM) na strani.*

Primeri: Dapper, Yahoo Pipes.

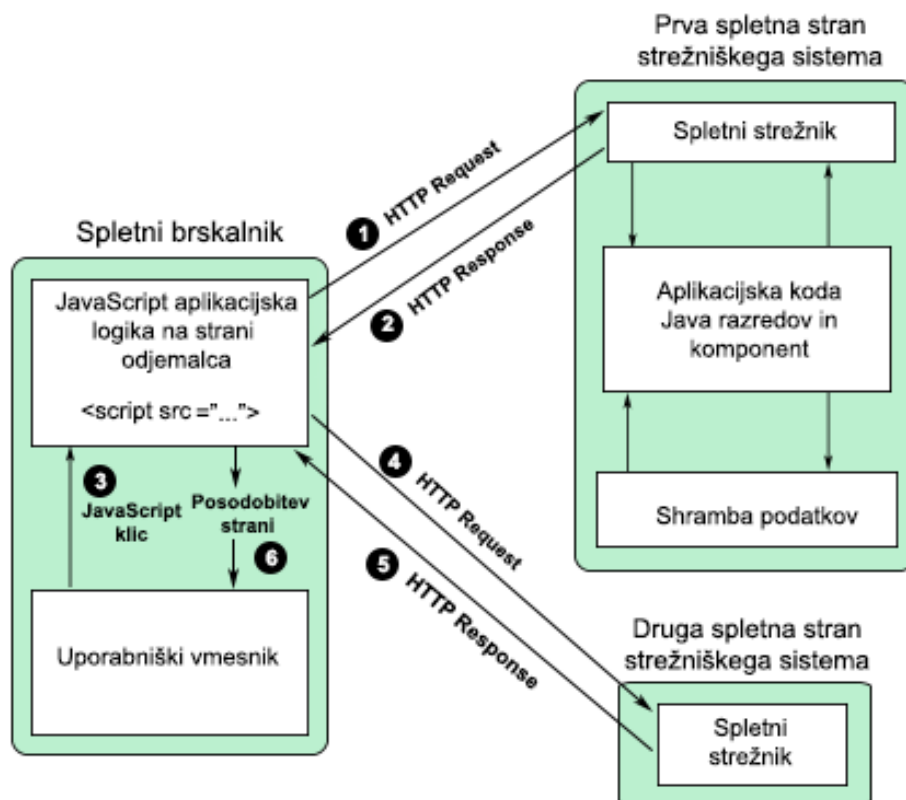
3.2.2 Prepletene storitve na strani odjemalca

Pri prepletenih storitvah na strani odjemalca se podatki in komponente integrirajo neposredno v odjemalčevem spletnem brskalniku. So enostavne za implementacijo, kajti v primeru, da spletna stran storitve, ki jo želimo vključiti, vsebuje knjižnico JavaScript, to enostavno vključimo na spletno stran prepletene storitve. Prednost pred prepletenimi storitvami na strežniku je, da je na strani odjemalca avtomatsko priskrbljena tudi strežniška komponenta.

Navadno so storitve na strani odjemalca tudi hitrejše, saj so zahtevki in odgovori avtomatsko posredovani med odjemalcem in strežnikom, kar ima za posledico krajši odzivni čas. Zaradi te neposredne poti je manjša tudi obremenitev strežnika, saj prenos ne vključuje posredniškega strežnika [27, 31].

Proces, ki opisuje delovanje prepletene storitve na strani odjemalca (glej sliko 3.3):

1. *Brskalnik pošlje zahtevo po spletni strani do strežnika.*
2. *Strežnik naloži spletno stran skupaj s knjižnicami JavaScript v brskalnik. Te knjižnice služijo za omogočanje spletne storitve (npr. s strani Google maps).*
3. *Iz brskalnika se kličejo funkcije iz knjižnice JavaScript. Te dinamično ustvarijo `<script>` element, ki kaže na stran prepletene storitve.*



Slika 3.3: Prepletena storitev na strani odjemalca [31]

4. Na osnovi elementa `<script>` je narejena zahteva po zagonu skripte na strani spletne storitve.
5. Stran prepletene storitve naloži skripto. Tipično se izvrši lokalna povratna funkcija v brskalniku s pomočjo JSON objekta.
6. Povratna funkcija posodobi odjemalčev pogled strani z manipulacijo DOM, ki je vsebovana na strani.

Storitve, ki uporabljajo kombinacijo obeh pristopov: WireCloud, RSSBus.

3.3 Namen prepletenih storitev

Prepletene storitve ločimo tudi glede na to, kakšni rabi so namenjene. Tako jih v prvi vrsti ločimo na: (1) *potrošniške* in (2) *poslovne*. Potrošniške so namenjene končnim uporabnikom, ki želijo razviti prepletene storitve v svoje lastne namene, medtem ko so poslovne namenjene predvsem podjetjem, ki implementirajo prepletene storitve za podporo poslovnim procesom. V nadaljevanju je podrobnejša analiza obeh omenjenih vrst.

3.3.1 Potrošniške

Te so namenjene uporabi posameznikov, da ustvarijo prepletene storitve za osebne namene, na primer za personalizirano spletno storitev. Potrošniške prepletene storitve združujejo elemente iz različnih virov s pomočjo grafičnih uporabniških vmesnikov. Namesto da uporabnik odpira množico spletnih strani, npr. za napoved vremena, novice in zemljevide, mu je dostop do teh storitev omogočen z individualne spletne strani. Te informacije jemljejo izključno s spleta prek protokolov RESTful, SOAP, RSS in Atom. Potrošniške prepletene storitve so dandanes tudi najbolj poznana oblika spletnih storitev [20, 22, 27].

Primeri potrošniških prepletenih aplikacij: Dapper, DERI pipes, Google Spreadsheets, Yahoo Pipes, Wirecloud.

3.3.2 Poslovne

Poslovne prepletene storitve so tiste, ki združujejo notranje vire, storitve in podatke iz zunanjih, javno dostopnih virov. Podatki so predstavljeni na enoten način, kar omogoča sodelovanje med podjetji in razvijalci. To je pomembno tudi za agilne razvojne projekte, kjer je za definiranje in implementacijo poslovnih zahtev bistveno sodelovanje med strankami in razvijalci.

Od potrošniških spletnih storitev se razlikujejo predvsem v tem, da te za-

gotavljajo tudi varnost, upravljanje, spremljanje in razpoložljivost [26].

Primeri poslovnih prepletenih aplikacij: Apatar, RSSBus, We Wired Web.

3.4 Ciljni uporabniki

Glede na zahtevnost pri razvoju prepletenih storitev, je vsaka izmed njih namenjena svoji skupini ciljnih uporabnikov. Nekatere zahtevajo predznanje različnih tehnologij in programiranja, medtem ko druge ne zahtevajo poznavanja skoraj nikakršnih tehničnih konceptov. Glede na to ločimo tri vrste ciljnih uporabnikov [25]:

1. **Razvijalec** je najbolj izkušen uporabnik z znanjem programiranja ter poznavanjem spletnih tehnologij, aplikacijskih programskih vmesnikov in drugih orodij za ustvarjanje prepletenih storitev.
2. **Napreden uporabnik** ne potrebuje programerskega znanja, mora pa dobro poznati specifično ogrodje za ustvarjanje prepletenih storitev.
3. **Neizkušen uporabnik** je uporabnik spleta, ki mora poznati le najbolj osnovne funkcionalnosti ogrodij za ustvarjanje prepletenih storitev.

Nekatera izmed ogrodij za razvoj prepletenih storitev ponujajo tudi dodatne funkcionalnosti, ki zahtevajo določeno znanje in veščine ter so tako lahko namenjeni več kot eni izmed omenjenih skupin uporabnikov.

Tabela 3.2 prikazuje kategorizacijo ogrodij glede na ciljne uporabnike. Kot je razvidno, ima večina ogrodij več skupin ciljnih uporabnikov. Opazimo, da je največ ogrodij namenjenih naprednim uporabnikom, ki za uporabo že potrebujejo določena znanja. Iz tabele prav tako lahko razberemo, katera ogrodja so namenjena več kot eni skupini uporabnikov. Ta pripadnost temelji na tem, kako napredno uporabo ogrodja želi sam uporabnik. Zahteva po razvijalcih

je potrebna predvsem pri razvoju lastnih komponent ali storitev (npr. Wire-Cloud).

ogrodje/uporabniki	razvijalci	napredni uporabniki	neizkušeni uporabniki
Apatar	x		
Dapper		x	x
DERI Pipes		x	
Google Spreadsheets	x	x	x
MarcoFlow	x		
Microsoft Excel	x	x	x
Yahoo Pipes		x	
YQL	x	x	
RSSBus	x		
Wirecloud	x	x	x
We Wired Web		x	x

Tabela 3.2: Kategorizacija ogrodij glede na ciljne uporabnike

3.5 Podpora uporabnikom in soustvarjanje vsebine

Ogrodja za ustvarjanje prepletenih storitev nudijo različne vrste podpore svojim uporabnikom. V veliko pomoč so navodila in primeri uporabe ogrodja ter podpora v obliki odpravljanja težav ali svetovanja. Nekateri ponudniki nudijo svojim uporabnikom tudi možnost izmenjave vsebin ali soustvarjanja komponent.

Glavne želene storitve, ki jih ogrodja nudijo, so:

- **Dokumentacija**

Dokumentacija, ki vsebuje podroben opis, je pomemben dejavnik, ki uporabniku olajša prvi stik z uporabo določenega ogrodja. Pomembna so tudi navodila za uporabo v obliki vaj.

- **Izmenjava vsebin in ponovna uporaba**

Možnost uporabnikov, da objavijo svoje prepletene storitve ali uporabijo že obstoječe.

- **Sodelovanje**

Predvsem za ciljno skupino razvijalcev lahko ogrodja nudijo možnost soustvarjanja novih komponent ali operatorjev ter njihovo objavo na spletni strani ogrodja.

- **Diskusija**

Diskusija se nanaša na obstoj skupnosti v obliki foruma, klepetalnice ali socialnih omrežij, kjer si lahko uporabniki izmenjujejo ideje in nasvete ter si nudijo medsebojno pomoč.

ogrodje/podpora	dokumentacija	izmenjava vsebin	sodelovanje	diskusija
Apatar	x	x	x	x
Dapper	x	x		
DERI pipes	x		x	
Google Spreadsheets	x		x	x
Microsoft Excel	x			x
RSSBus	x			x
Yahoo Pipes	x	x		x
YQL	x			x
We Wired Web	x		x	x
Wirecloud	x	x	x	x

Tabela 3.3: Kategorizacija ogrodij glede na podporo uporabnikom in soustvarjanje vsebine

V tabeli 3.3 je prikazano, katere od naštetih storitev za podporo uporabnikom in soustvarjanje vsebine nudijo posamezna ogrodja. Opazimo, da večina ogrodij nudi dokumentacijo, najmanj pa izmenjavo vsebin.

Poglavje 4

Pregled ključnih ogrodij za razvoj podatkovno usmerjenih aplikacij

To poglavje je namenjeno pregledu ogrodij za razvoj prepletenih aplikacij. Podrobneje je raziskanih pet in za vsakega izmed njih je narejen splošen opis, kjer je opisan njihov namen, sestava in delovanje.

Glavni kriterij za izbor ogrodij, predstavljениh v tem poglavju, je njihova raznolikost. Zajeti smo želeli ogrodja z različno zahtevnostjo uporabe in različnim namenom. Pozorni smo bili tudi na sam način razvoja prepletene storitve znotraj uporabniškega vmesnika, ki se razlikuje tudi po številu in lastnostnih razpoložljivih komponent, operatorjev in storitev. Na podlagi omenjenih kriterijev smo se odločili za podrobnejšo predstavitev predstavnikov *Apatar*, *Dapper*, *Google Spreadsheets*, *WireCloud* in *Yahoo Pipes*.

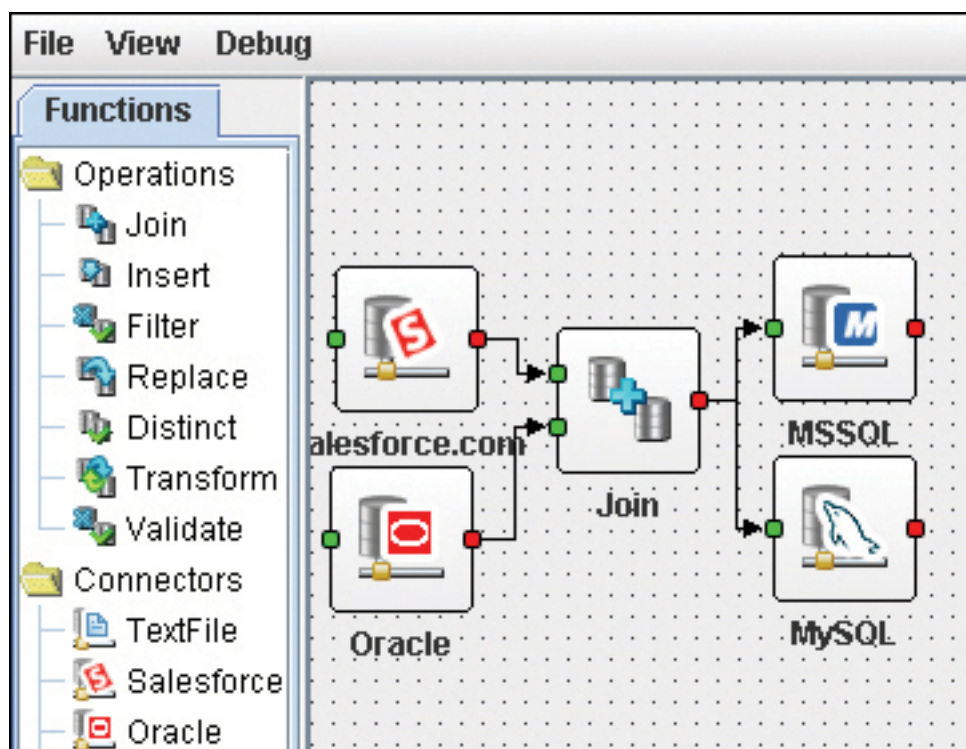
4.1 Apatar

Apatar [2] je odprtokodna programska rešitev za integracijo podatkov. Ima široko zbirko operacij ETL, ki omogočajo uporabnikom povezavo konektorjev

s pomočjo različnih operatorjev.

Ti konektorji so:

- **podatkovne baze** (npr. MySQL, MS SQL, Oracle, PostgreSQL)
- **aplikacijske rešitve** (npr. Salesforce.com, SugarCRM, QuickBooks)
- **aplikacije iz Spleta 2.0** (npr. Flickr, Amazon)
- **strukturirane/nestrukturirane datoteke** (npr. XML, CSV, TXT)



Slika 4.1: Prikaz delovnega okolja Apatar

Uporabniki skozi preslikave na grafičnem uporabniškem vmesniku izvajajo želene transformacije. Pri tem ni potrebno znanje programiranja, tudi za izvajanje naprednejših integracij ne. Apatar se navadno uporablja z namenom

podatkovnega skladiščenja, migracije podatkov, sinhronizacije in aplikacijske integracije. Uporabljen programski jezik je Java J2EE v razvijalnem okolju Eclipse. Uporabljeni jeziki za delanje poizvedb pa so SQL in SOAP/XML.

Glavne prednosti Apatar ogrodja so naslednje: zmanjšanje stroškov integracije, izboljšana kakovost podatkov, krajši implementacijski čas, nižji stroški vzdrževanja in odprava programiranja.

4.2 Dapper

Dapper [5] je brezplačna spletna storitev, ki omogoča zajem in uporabo podatkov s spletnih strani. Ta zajeta vsebina se imenuje Dapp in je lahko pridobljena brez programerskega znanja. V primeru, da ima uporabnik potrebne programerske veščine, lahko Dapp vključi tudi v spletne aplikacije ali strani ter avtomatizira proces pridobivanja Dapp vsebin za več različnih spletnih strani.

Pri kreiranju prepletenih storitev so lahko razvite lastne storitve ali uporabljene tiste, ki so jih ustvarili drugi uporabniki. V nadaljevanju bo podrobneje opisana prva možnost. Prepletena storitev je ustvarjena znotraj Dapp tovarne, kjer se nahaja možnost vnosa spletne povezave. Na tej lahko označimo, točno katere podatke želimo zajeti. Nato izberemo tako imenovani format, v katerega bodo ti pretvorjeni. Pri tem lahko izbiramo med šestimi različnimi opcijami:

- **Naročnine** (vir RSS, filtriran vir RSS, vir Atom))
- **Vgrajevanje v spletno stran** (Flash Widget)
- **Vstavljanje v začetno stran** (Google Gadget, Netvibes Module, FageFlake)
- **Vstavljanje v zemljevid** (Google Maps)
- **Vstavljanje v koledar** (iCal)
- **Za gradnjo** (XML, JSON, XSL, YAML, HTML, EMAIL)

Where is the content you want to use?

In a website

In an existing RSS Feed

Enter the URL of the website:

[Advanced](#)

Choose a format (You can always change this later)



- ☐ Dapp XML
- ☒ **RSS feed**
- ☐ Google Gadget
- ☐ Netvibes Module
- ☐ Google Map
- ☐ iCalendar
- ☐ Image Loop

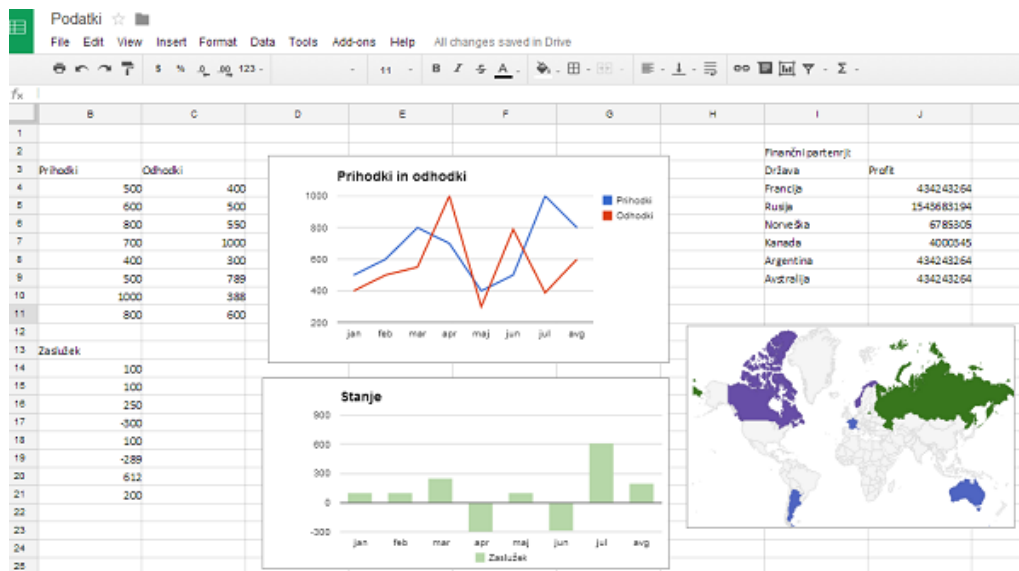
Slika 4.2: Prikaz delovnega okolja Dapper

Dapper je tudi repozitorij, saj ponuja spletno skupnost, kjer lahko uporabniki delijo svoje prepletene aplikacije.

4.3 Google Spreadsheets

Google Spreadsheets [7] je spletna storitev ponudnika Google, ki je v osnovi namenjena ustvarjanju, posodabljanju in spreminjanju tabel znotraj spletnega brskalnika in njihovemu deljenju na spletu. Temelji na tehniki Ajax in ima enostaven uporabniški vmesnik. Tega sestavljajo menijska vrstica, vnosno polje

za vnašanje funkcij ter delovna površina sestavljena iz celic, vrstic in stolpcev.



Slika 4.3: Prikaz delovnega okolja Google Spreadsheets

Spodaj je naštetih nekaj funkcionalnosti, ki jih storitev ponuja:

- **Operacije nad podatki** kot so na primer razvrščanje, filtriranje, pogledi in pivotne tabele.
- **Ustvarjanje grafikonov** kot so krožni, črtni, stolpčni, radarski in drugi.
- **Razširitve** (možnost razširitve z Apps Script ali dodatki, razvitimi s strani tretjih ponudnikov).

Za primer gradnje interaktivne prepletene storitve znotraj Google Spreadsheets je priporočljiva uporaba storitve Apps Script, za katero mora imeti uporabnik tudi programerske veščine.

4.4 WireCloud

WireCloud [14] je odprtokodna rešitev, ponujena v okviru projekta FI-WARE. V prvi vrsti uporabniki združujejo komponente čelnega dela sistema z na-

menom ustvarjanja prepletenih storitev. Te komponente lahko izbiramo iz kataloga in jih med seboj povezujemo na grafičnem vmesniku. Druga vrsta uporabnikov so razvijalci, ki jim je ponujena možnost programiranja novih komponent in operatorjev ter njihovo dodajanje v katalog.

Glavni orodji, ki ju WireCloud ponuja:

- **Urejevalnik povezav** (ang. *wiring editor*)

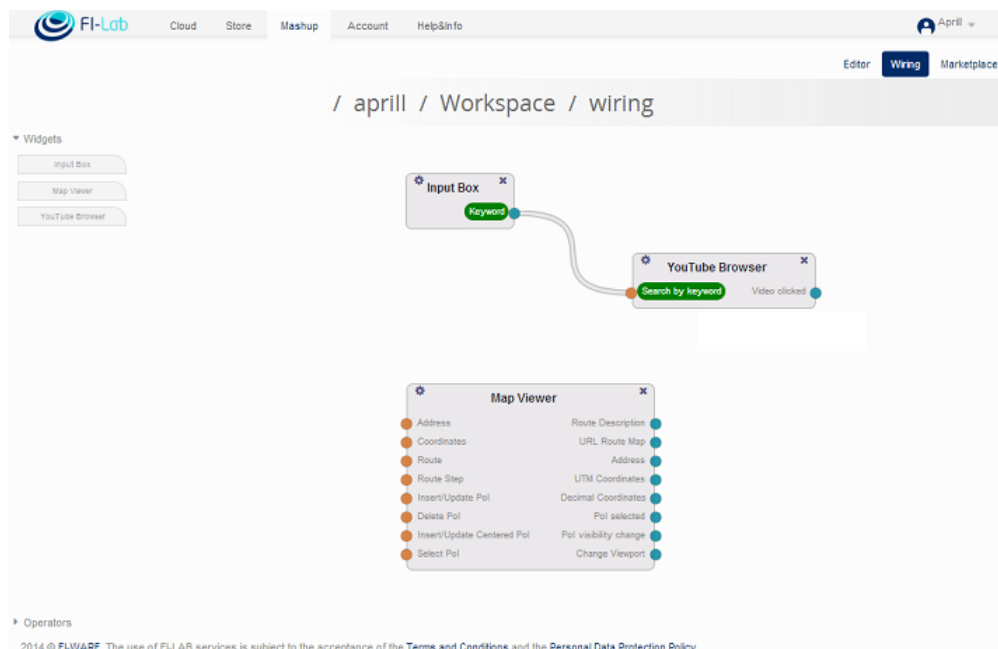
Ta omogoča povezovanje gradnikov v prepleteni storitvi z namenom, da se ustvari bogata spletna aplikacija.

- **Cevovodni urejevalnik** (ang. *pipeline editor*)

V tem urejevalniku je mogoče povezati gradnike s storitvami zalednega dela sistema ali s podatkovnimi viri s pomočjo zbirke operatorjev (npr. filtracijo, agregacijo).

WireCloud tako integrira podatke, aplikacijsko logiko in komponente uporabniškega vmesnika. Njegova glavna značilnost je hiter razvoj po principu »naredi sam« (DIY), ki ne zahteva programerskega znanja. Uporabljene tehnologije na strani odjemalca so JavaScript, HTML in CSS, na strani strežnika pa se uporablja Python.

Platforma WireCloud je zgrajena iz treh ključnih komponent: je zgrajena iz treh ključnih komponent: (1) *aplikacijskega urejevalnika prepletenih storitev*, (2) *poganjevalnika prepletenih storitev* in (3) *kataloga*. Aplikacijski urejevalnik prepletenih storitev je spletni urejevalnik, ki končnim uporabnikom omogoči kreiranje prepletenih storitev. Sestavljajo ga delovno okolje, kamor lahko uporabnik postavlja komponente, orodje za njihovo povezovanje in katalog, kjer lahko dostopa do zelenih komponent ali storitev. Poganjevalnik prepletenih storitev nudi povezovanje komponent, obstojnost stanj in povezovanje naprav namestniškega strežnika od programskega vmesnika do urejevalnika. Izvajanje prepletene storitve zahteva od poganjevalnika, da koordinira izvajanje komponent in njihovo medsebojno komunikacijo. Razširjen programski vmesnik



Slika 4.4: Prikaz delovnega okolja WireCloud

ponuja dodatne module, ki jih lahko dodamo poganjevalniku (npr. moduli za varnost, objavo, naročnino ali cevovodni moduli).

4.5 Yahoo Pipes

Yahoo Pipes [15] je spletna aplikacija, ki ponuja grafični uporabniški vmesnik za gradnjo prepletenih storitev, ki združujejo spletne vire, spletne strani in druge storitve ter iz njih ustvarijo aplikacijo. Ta deluje tako, da omogoči pridobivanje podatkov s pomočjo cevi in vzpostavitev pravil, kako je ta vsebina obdelana (npr. filtracija).

Cev je del cevovoda za obdelavo podatkov, ki združuje enega ali več podatkovnih virov in operatorje za njihovo manipulacijo (npr. sortiranje ali filtriranje) ter dodatne funkcije (npr. zanke, regularne izraze, števce) [33].

Yahoo Pipes vsebuje tri glavna podokna:

- **Knjižnico**

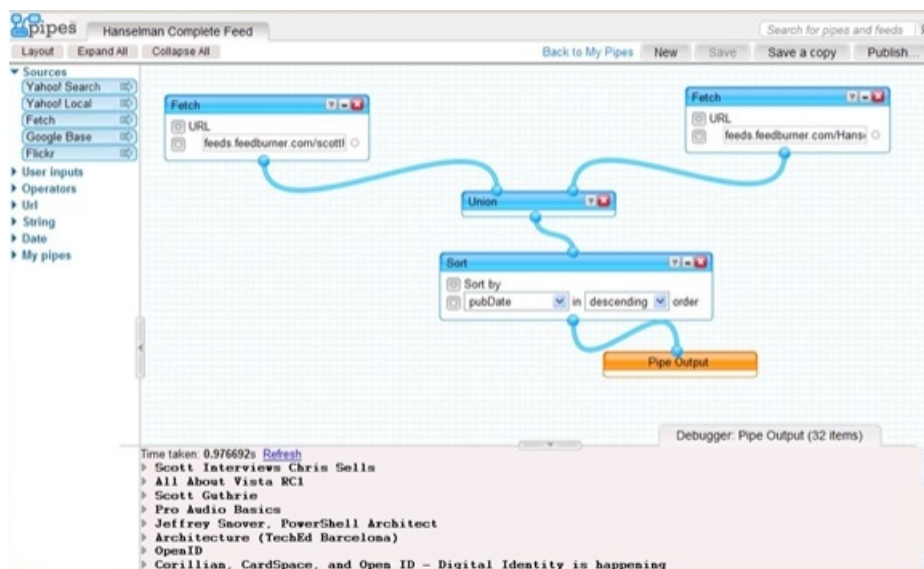
V knjižnici so zbrani moduli, ki jih lahko uporabnik izbere in povleče na risalno površino. Posamezni moduli so združeni v naslednje skupine: (1) *viri*, (2) *uporabniški vhodi*, (3) *operatorji*, (4) *URL-ji*, (5) *nizi*, (6) *datumi*, (7) *lokacije* in (8) *številke*.

- **Risalna površina**

Glavna površina, kjer poteka urejanje cevovoda. Nanjo je mogoče vstaviti izbrane module iz knjižnice in jih medsebojno povezati.

- **Razhroščevalnik**

Med urejanjem cevovoda lahko uporabnik izbere modul na risalni plošči, razhroščevalnik pa pokaže njegov izhod.



Slika 4.5: Prikaz delovnega okolja Yahoo pipes

Poglavje 5

Večkriterijski odločitveni model za izbiro ogrodja za razvoj podatkovno usmerjenih aplikacij

To poglavje je namenjeno izgradnji večkriterijskega odločitvenega modela na podlagi izbranih ogrodij, opisanih v prejšnjem poglavju. Najprej so ta ocenjena glede na izbrane kriterije, čemur sledi še njihova primerjava. Prvi del poglavja je namenjen podrobnejšemu opisu kriterijev za ocenjevanje, skupaj z njihovimi podkriteriji in zalogami vrednosti. V drugem delu so prikazani rezultati in končne ugotovitve ter rangiranje izbranih ogrodij.

Za začetek smo izdelali odločitveno drevo, ki se deli na tri glavne kriterije (glej sliko 5.1). Za gradnjo večkriterijskega odločitvenega modela je uporabljen program Dexi [1].



Slika 5.1: Odločitveno drevo

5.1 Opis kriterijev

Pri gradnji odločitvenega modela smo se osredotočili na tri glavne kriterije, ki so navedeni v tabeli 5.1. Vsakemu izmed njih smo določili podkriterije ter odvisno od njih določili funkcijo koristnosti. Tako je na najvišjem nivoju vsak izmed glavnih kriterijev ocenjen z eno izmed treh oziroma štirih vrednosti, prav tako razvidnih iz spodnje tabele. Razlog, da smo se pri kriteriju funkcionalnosti odločili za večjo zalogo vrednosti, je, da ima slednji več podkriterijev v primerjavi z drugima dvema. Na podlagi tega da tovrstna kategorizacija natančnejšo analizo.

Uporabniška izkušnja	<i>slaba</i>	<i>zadovoljiva</i>	<i>dobra</i>	
Podpora uporabnikom	<i>slaba</i>	<i>zadovoljiva</i>	<i>dobra</i>	
Funkcionalnost	<i>slaba</i>	<i>zadovoljiva</i>	<i>dobra</i>	<i>odlična</i>

Tabela 5.1: Glavni kriteriji

5.1.1 Uporabniška izkušnja

Kriterij opisuje izkušnjo, ki jo ima uporabnik ob uporabi določenega ogrodja. Osredotočili smo se na naslednje podkriterije, ki vplivajo na njegovo pozitivno ali negativno izkušnjo:

1. **Enostavnost uporabe** je nivo težavnosti, s katerim se uporabnik sooči ob uporabi določenega ogrodja. Ker je predznanje uporabnika dejavnik, ki ima močan vpliv na ta podkriterij, smo predpostavili, da gre za uporabnika brez večjega predhodnega znanja.
2. **Tekoče delovanje** opisuje oteženost oz. neoteženost delovanja programa. Na to vpliva dejavnik pogostosti pojavljanja težav, kot so na primer pojav napak ob uporabi ali prenehanje delovanja ogrodja.
3. **Hitrost** opisuje, kako hiter je razvoj prepletene storitve s pomočjo določenega ogrodja. Pri tem je vsebovana tudi hitrost v primerih velike obremenitve.

Tabela 5.2 prikazuje omenjene podkriterije uporabniške izkušnje ter njihove zaloge vrednosti.

Enostavnost uporabe	<i>enostavna</i>	<i>srednje zahtevna</i>	<i>zahtevna</i>
Tekoče delovanje	<i>močno oteženo</i>	<i>občasno oteženo</i>	<i>neoteženo</i>
Hitrost	<i>nizka</i>	<i>srednja</i>	<i>visoka</i>

Tabela 5.2: Zaloge vrednosti podkriterijev uporabniške izkušnje

5.1.2 Podpora uporabnikom

Pri tem kriteriju smo se osredotočili na dejavnike, ki nudijo podporo uporabnikom posameznih ogrodij:

1. **Navodila** zajemajo dokumentacijo in primere uporabe ogrodja v besedilni, slikovni ali video obliki.

2. **Diskusija** se nanaša na obstoj forumov, klepetalnic ali socialnih omrežij, namenjenim diskusiji o specifičnem ogroddju in njegovi uporabi.
3. **Spletna stran** opisuje predvsem kakovost spletne strani, glede na kriterije kot so preglednost, aktualnost, vsebina, izgled in delovanje.
4. **Odpravljanje težav** je možnost ponudnika za reševanje uporabniških težav pri uporabi ogrodja.

V tabeli 5.3 so ponovno naštetii podkriteriji podpore uporabnikom skupaj z njihovo zalogo vrednosti.

Navodila	<i>brez</i>	<i>osnovna</i>	<i>odlična</i>
Diskusija	<i>brez</i>	<i>osnovna</i>	<i>odlična</i>
Spletna stran	<i>slaba</i>	<i>osnovna</i>	<i>odlična</i>
Odpravljanje težav	<i>brez</i>	<i>osnovno</i>	<i>odlično</i>

Tabela 5.3: Zaloga vrednosti podkriterijev podpore uporabnikom

5.1.3 Funkcionalnost

Kriterij opisuje, ali posamezno ogrodje podpira določene funkcionalnosti, ki so v tem modelu predstavljene kot podkriteriji:

1. **Razširljivost** je dodatna funkcionalnost, ki je v prvi vrsti namenjena predvsem razvijalcem. Zajema ustvarjanje novih komponent in operatorjev ter njihovo objavo na strani ali dodajanje v katalog.
2. **Iskanje po bazi** je možnost iskanja prepletenih storitev, razvitih s strani drugih uporabnikov, in njihovo ponovno uporabo. V kolikor je ponujeno samo iskanje po bazi, je funkcionalnost ocenjena kot *osnovna*, v kolikor pa ponuja še ponovno uporabo pa kot *napredna*.

3. **Razpoložljivost orodij** opisuje število že obstoječih komponent, konektorjev, operatorjev ali programskih vmesnikov za razvoj prepletenih storitev specifičnega ogrodja.
4. **Integracija podatkov** je možnost uporabe lastnih podatkov v obliki podatkovnih baz, Excelovih datotek, sistemov ERP in drugih ter njihove integracije znotraj prepletenih storitev.
5. **Delno strukturirani viri** imenujemo funkcionalnost, ki opisuje, ali ogrodje podpira integracijo delno strukturiranih virov, kot sta na primer XML in JSON, pri procesu razvoja prepletenih storitev.
6. **Storitve za souporabo** opisujejo možnost vključevanja spletno dostopnih storitev, namenjenih souporabi (npr. Flickr, Dropbox) v prepleteno storitev.

V tabeli 5.4 so predstavljeni podkriteriji funkcionalnosti in njihove zaloge vrednosti.

Razširljivost	<i>brez</i>	<i>osnovna</i>	<i>napredna</i>
Iskanje po bazi	<i>brez</i>	<i>osnovno</i>	<i>napredno</i>
Razpoložljivost orodij	<i>majhna</i>	<i>srednja</i>	<i>velika</i>
Integracija podatkov	<i>brez</i>	<i>omejena</i>	<i>napredna</i>
Delno strukturirani viri	<i>nepodprti</i>	<i>delno podprti</i>	<i>podprti</i>
Storitve za souporabo	<i>nepodprte</i>	<i>delno podprte</i>	<i>podprte</i>

Tabela 5.4: Zaloga vrednosti podkriterijev funkcionalnosti

5.1.4 Ocenjevanje

Vsi podkriteriji imajo zalogo vrednosti sestavljeno iz treh različno ovrednotenih vrednosti. Tako v sklopu ocenjevanja posameznega kriterija določimo številske vrednosti podkriterijev, ki se nato preslikajo v zaloge vrednosti na najvišji

ravni.

Če podkriterij zavzema:

- *slabo vrednost*, je ocenjen z 0 točkami;
- *neutrarno vrednost*, je ocenjen z 0,5 točke;
- *dobro vrednost*, je ocenjen z 1 točko.

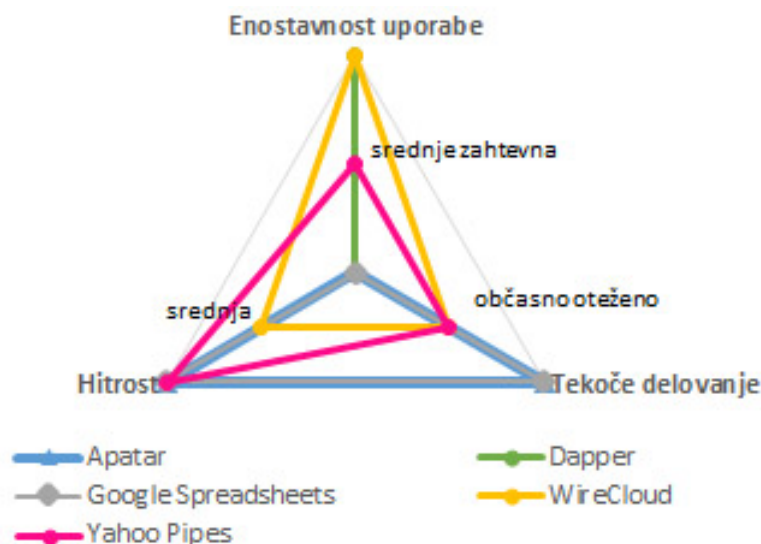
Za prva dva kriterija, **uporabniška izkušnja** in **podpora uporabnikom**, veljajo enaka odločitvena pravila. Če je vrednost vsote od 0 do vključno 1 točke, se preslika v vrednost *slaba*. Če šteje več kot 1 točko do vključno 2 točki, zavzame vrednost *zadovoljiva*. Če ima več kot 2 točki, se preslika v vrednost *dobra*.

Za kriterij **funkcionalnost** veljajo drugačna pravila, saj se vsota ovrednotenih podkriterijev lahko preslika v štiri različne vrednosti. Če se vrednost giblje od 0 do vključno 1,5 točke, se preslika v vrednost *slaba*. Za intervale večje od 1,5 točke do vključno 3 točk zavzame vrednost *zadovoljiva*. Če šteje več kot 3 točke do vključno 4,5 točk, se preslika v vrednost *dobra*. V kolikor zavzema več kot 4,5 točke pa zavzame vrednost *odlična*.

5.2 Rezultati

V tej sekciji so predstavljeni rezultati večkriterijskega odločitvenega modela prepletenih aplikacij. Prvi del prikazuje rezultate vsakega izmed treh kriterijev, neodvisnih drug od drugega ter predlaga najboljšo izbiro ogrodja. V drugem delu je predstavljen končni radarski diagram glede na vse tri kriterije. Na osnovi tega je nato narejeno rangiranje posameznih ogrodi.

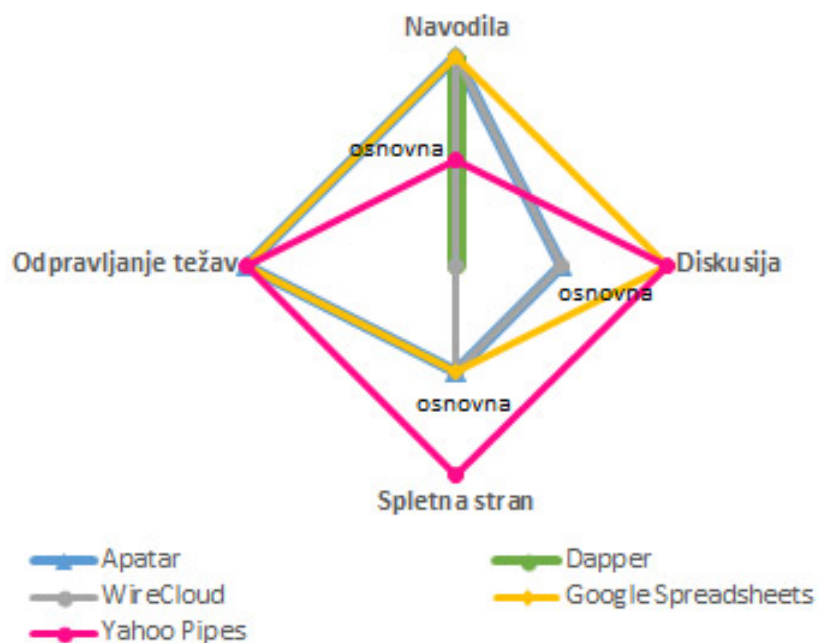
Graf na sliki 5.2 prikazuje ogrodja glede na njihovo uporabniško izkušnjo. Opazimo, da je najslabša izbira ogrodje Dapper, saj je kljub svoji enostavnosti počasno s pogostim pojavom različnih težav. Boljše izbire so Apatar, Google



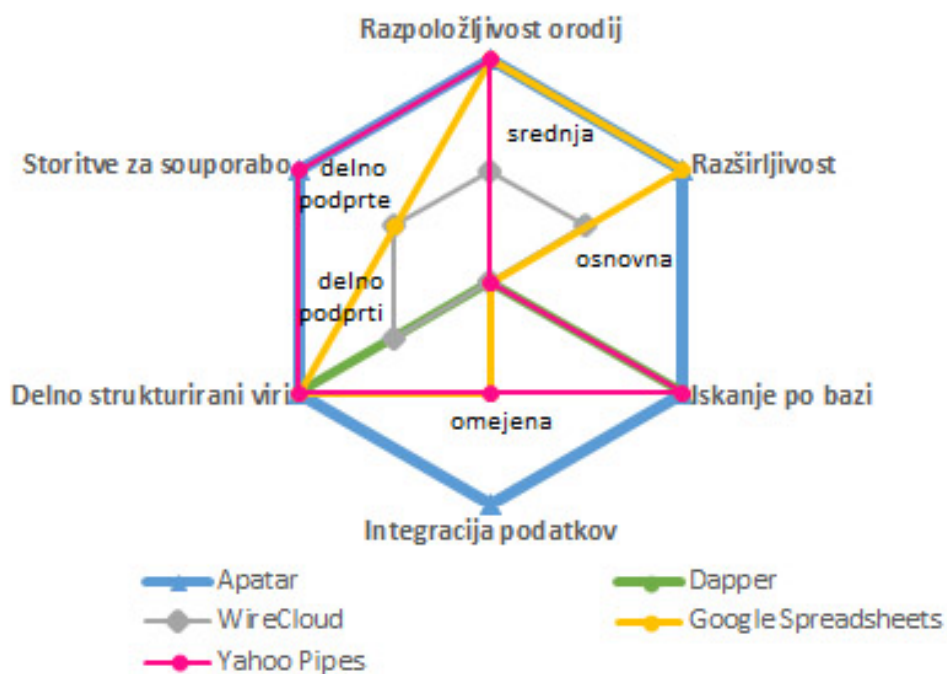
Slika 5.2: Radarski diagram uporabniške izkušnje

Spreadsheets, WireCloud in Yahoo Pipes. Kateri izmed njih je najboljši, je odvisno od izkušenosti uporabnika. Če gre za začetnika, je zanj najboljša izbira WireCloud. V primeru, da gre za naprednejšega uporabnika, pa mu svetujemo ogrodja Apatar ali Google Spreadsheets, ki se odlikujeta predvsem po svoji hitrosti in tekočem delovanju.

Naslednji graf (glej sliko 5.3) prikazuje podporo uporabnikom glede na štiri izbrane podkriterije. Na prvem mestu je Yahoo Pipes, ki ima prednosti pri odpravljanju težav, možnosti diskusije in spletne strani. Izenačeno z njim je tudi ogrodje Google Spreadsheets, ki nudi boljša navodila kot prejšnje ogrodje, a ima slabšo spletno stran. Na zadnjem mestu z najslabšo podporo uporabnikom je ogrodje Dapper, ki v pozitivnem smislu izstopa le zaradi odličnih navodil in primerov uporabe.

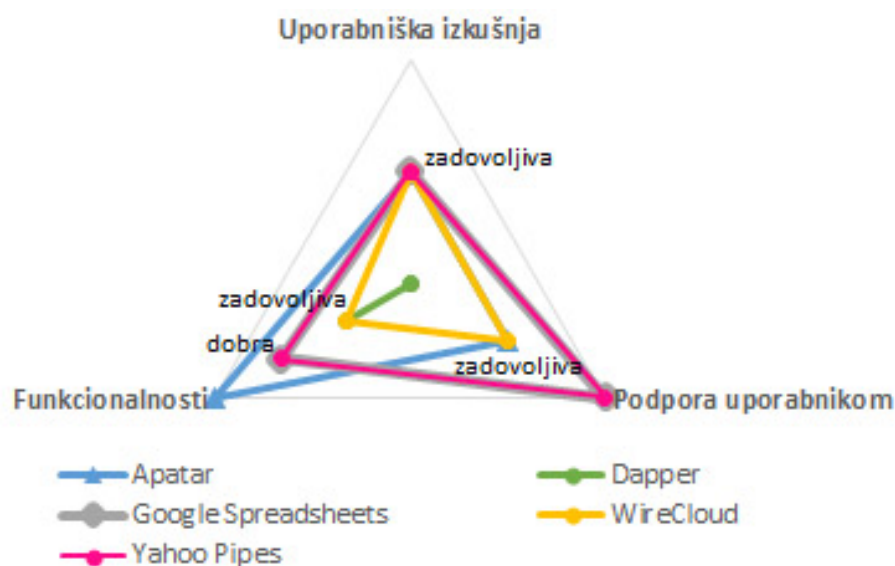


Slika 5.3: Radarski diagram podpore uporabnikom



Slika 5.4: Radarski diagram funkcionalnosti

Radarski graf 5.4 prikazuje ogrodja glede na kriterij funkcionalnosti. Razvidno je, da se tukaj najboljše obnese Apatar, ki zavzema najboljše vrednosti pri vseh šestih podkriterijih. Sledi mu Yahoo Pipes, ki je brez razširljivosti in z omejeno integracijo lastnih podatkov. Na tretjem mestu je Google Spreadsheets, ki nima podprte možnosti vključitve delno strukturiranih virov in iskanja po bazi. Na predzadnjem mestu je WireCloud brez možnosti integracije podatkov in iskanja po bazi. Najslabše se obnese ogrodje Dapper, ki podpira le delno strukturirane vire in možnost iskanja po bazi.



Slika 5.5: Rezultat: Radarski diagram treh glavnih kriterijev

V končni primerjavi vseh treh glavnih kriterijev (uporabniške izkušnje, podpore uporabnikom in funkcionalnosti) lahko iz grafa 5.5 ugotovimo, da sta najboljše ocenjeni ogrodji Yahoo Pipes in Google Spreadsheets, ki imata odlično podporo uporabnikom, dobro podprte funkcionalnosti in zadovoljivo uporabniško izkušnjo. Tesno jima sledi Apatar, ki se bolje obnese glede funkcionalnosti, nudi pa malce slabšo podporo uporabnikom. Na tretjem mestu je ogrodje WireCloud z vsemi tremi kriteriji ocenjenimi kot zadovoljivimi. Na zadnjem mestu je ponovno Dapper, ki ima prav tako kot prejšnje ogrodje svoje

funkcionalnosti ocenjene kot zadovoljive, vendar ima nekoliko slabšo podporo uporabnikom in uporabniško izkušnjo.

Poglavje 6

Implementacija prepletene storitve

To poglavje je namenjeno opisu izdelave prepletene storitve. V ta namen sta izbrana Google Spreadsheets in Yahoo Pipes, ki sta se glede na rezultate večkriterijskega odločitvenega modela iz prejšnjega poglavja izkazala kot najbolj ocenjeni ogrodji.

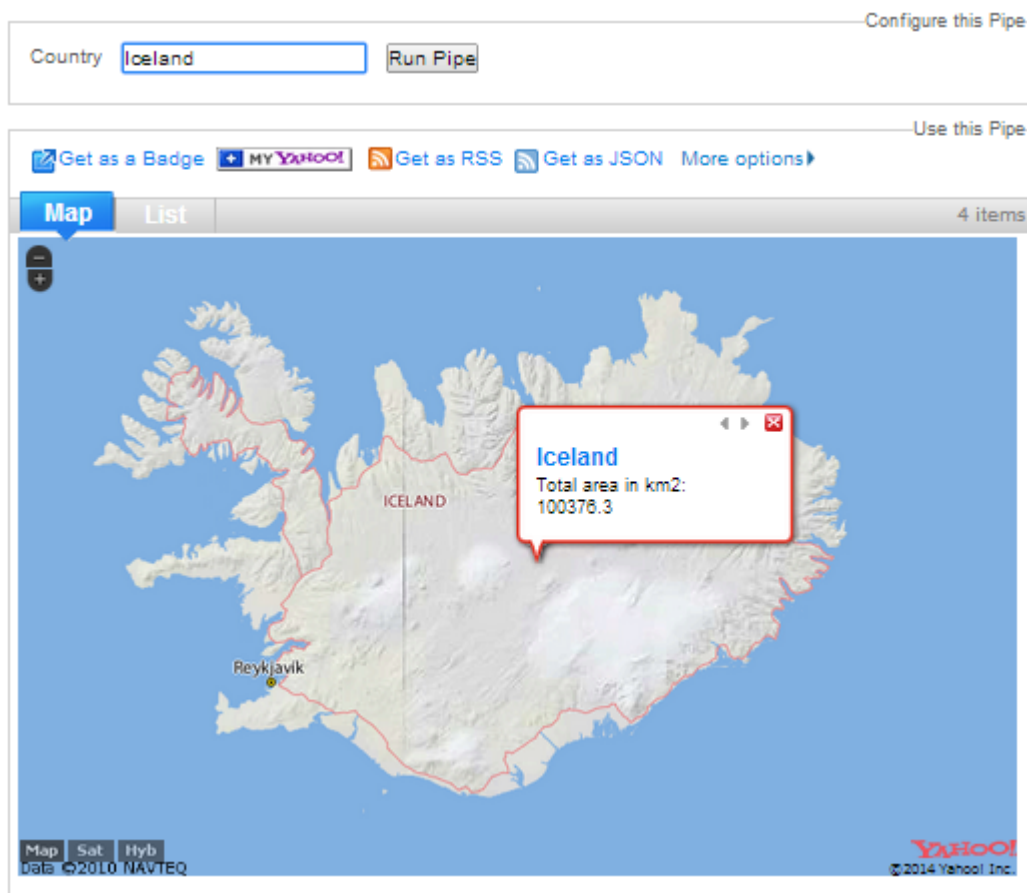
Izbrali smo si problemsko domeno, ki je namenjena tistim, ki radi potujejo in želijo o določeni državi izvedeti več. Prepletena storitev ponuja možnost prikaza izbrane države na zemljevidu ter o njej nudi dodatne informacije v obliki podatkovnih virov.

6.1 Prepletena storitev v ogrodju Yahoo Pipes

Podpoglavje opisuje razvoj prepletene storitve v ogrodju Yahoo Pipes po korakih. Najprej je narejen splošen opis z realiziranimi funkcionalnostmi, nato pa sledi še podrobnejša razlaga posameznih delov razvoja.

6.1.1 Opis delovanja prepletene storitve

Uporabnik najprej vnese ime države v vnosno polje. Ob pritisku na gumb se uporabniku prikaže okno z dvema zavihkoma. Prvi zavihek prikaže izbrano destinacijo na zemljevidu skupaj z oblačkom, v katerem je nekaj dodatnih informacij o njej. Slednje prikazuje slika 6.1.

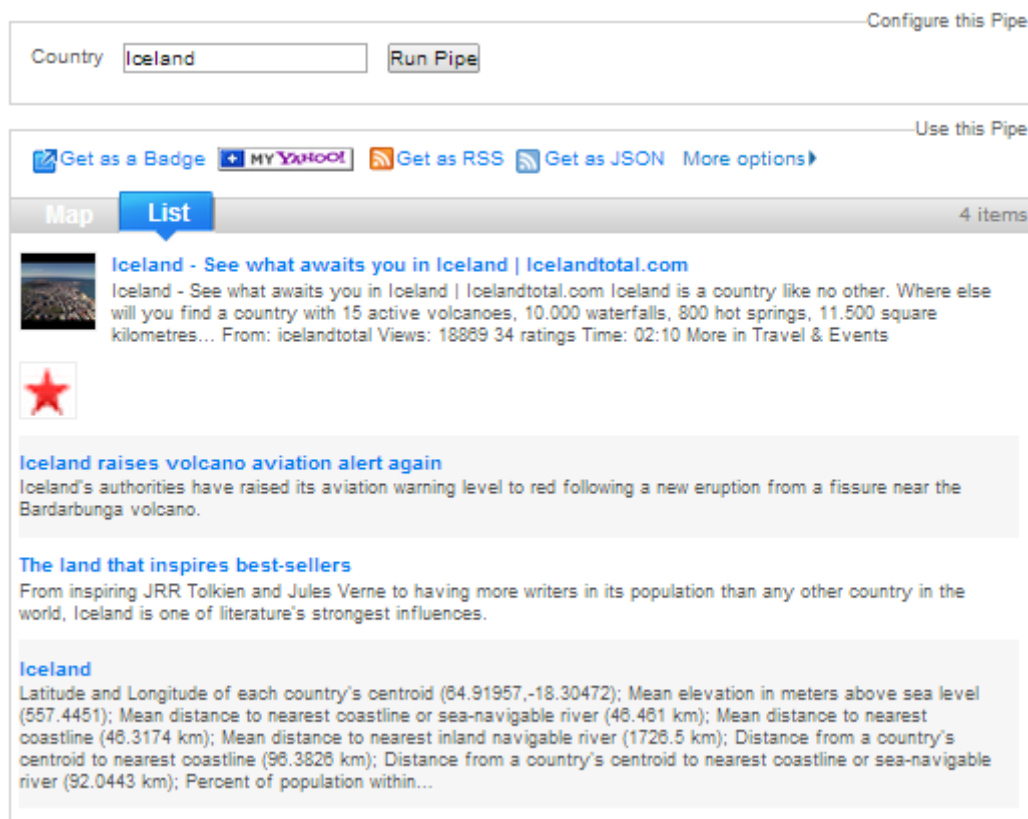


Slika 6.1: Zemljevid z dodatnimi informacijami o izbrani državi

Ob pritisku na ime države v oblačku se odpre novo okno z zapisom z Wikipedije [13] o izbrani državi.

Ob pritisku na drugi zavihek se odpre zbirka virov z dodatnimi informacijami o državi, kot prikazuje slika 6.2. Ti viri so videoposnetki z YouTube [16],

aktualne novice s strani Breaking Travel News [4] ter potovalni nasveti s strani BBC Travel[3].



Slika 6.2: Zbirka virov o izbrani državi

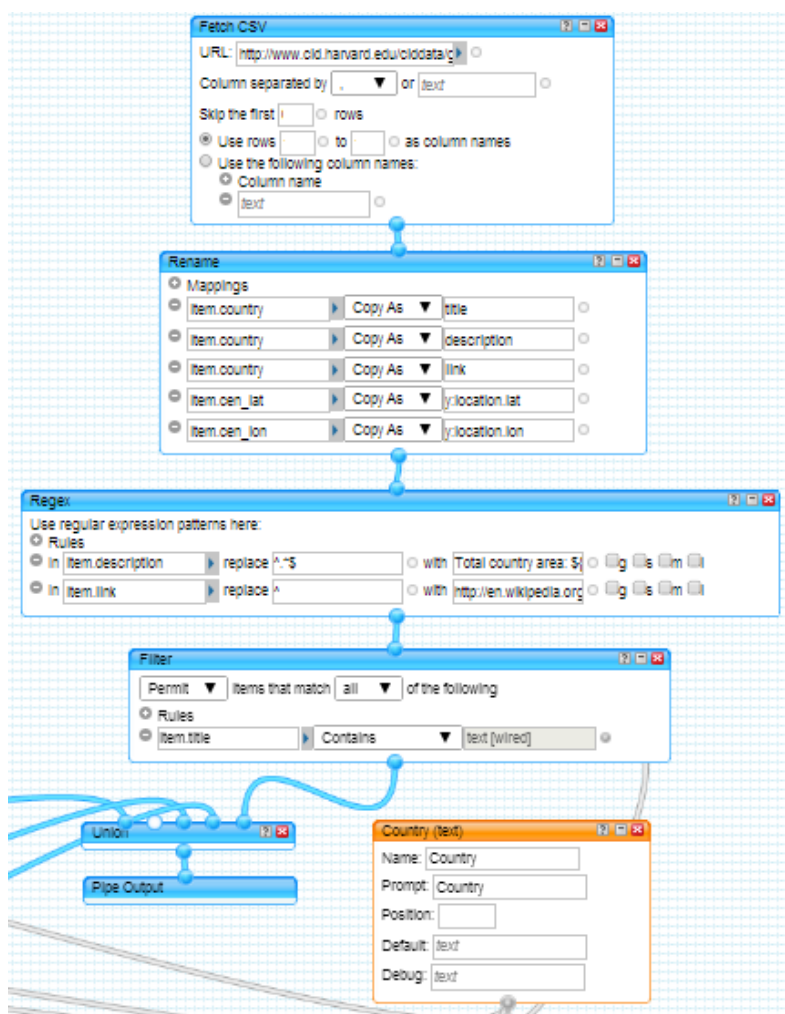
6.1.2 Izdelava prepletene storitve

V tej sekciji bo podrobneje prikazan razvoj prepletene storitve po delih.

Prikaz zemljevida

Za začetek bo podrobneje razložen del storitve iz slike 6.3. S pomočjo komponente *Fetch CSV* smo pridobili podatke o državah iz Harvardske podatkovne zbirke [23]. Povezali smo jo s komponento *Rename*, ki izbranim ele-

mentom iz podatkov ustvari duplikate pod novim imenom. Vrednosti iz spremenljivk `item.cen_lat` in `item.cen_lon` kopiramo v privzet tip spremenljivk `y:location`, iz katerih nato Yahoo Pipes prebere koordinate posameznih držav in jih označi na zemljevidu. Povezava vodi do *Regex* modula, ki ima možnost



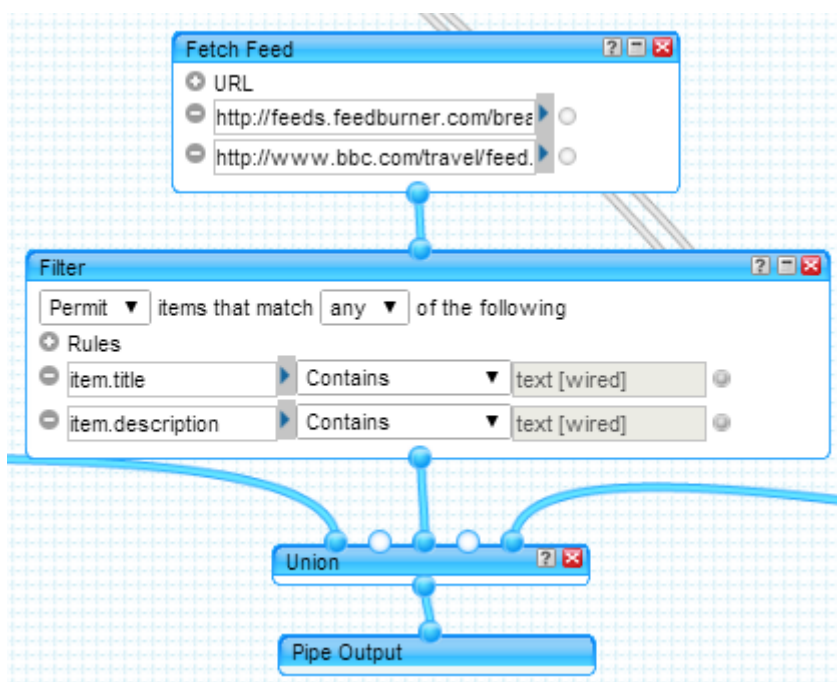
Slika 6.3: Del storitve za prikaz zemljevida

uporabe regularnih izrazov. Z njim zamenjamo `item.description` s podatkom o številu prebivalcev države, ki ga podamo v obliki `${areakm2}`. Vrednost elementa `item.link` pa zamenjamo s povezavo do Wikipedije, ki v storitvi

omogoči, da se ob pritisku na ime države odpre novo okno z zapisom Wiki o destinaciji. Komponenta *Filter* je povezana z vnosnim poljem (*Text Input*) in na ta način dovoljuje le prikaz podatkov, v katerem naslov (v našem primeru je to ime države) vsebuje vrednost vnosnega polja.

Pridobivanje podatkovnih virov

Aktualne informacije iz podatkovnih virov so zbrane s pomočjo komponente *Fetch Feed*, v katero so vnesene povezave do spletnih strani v obliki virov RSS. Primer prikazuje slika 6.4.

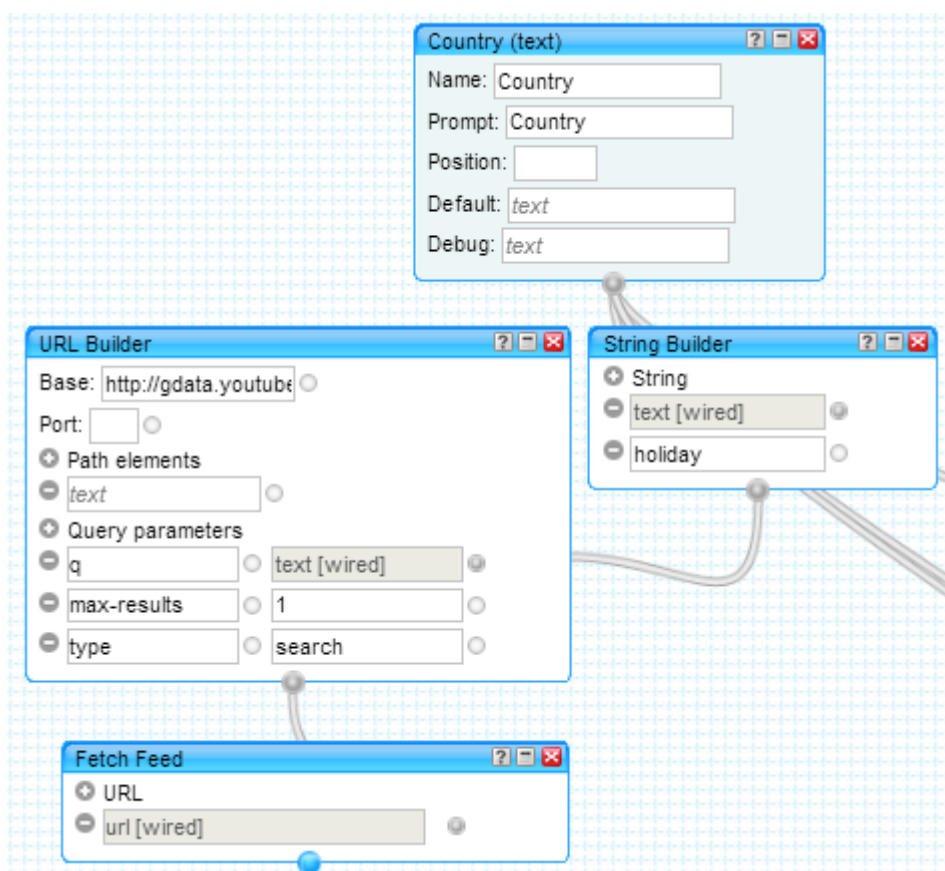


Slika 6.4: Pridobivanje podatkovnih virov

6.1.3 Videoposnetki

Ta korak opisuje pridobitev videoposnetkov iz storitve YouTube. Pri tem smo uporabili komponento *URL builder*, kjer poleg povezave vnesemo še iskalne

parametre. Omejili smo se na iskanje največ enega videoposnetka. Iskalno geslo q je sestavljeno iz dveh delov. Prvi del sestavlja država iz vnosnega polja, povezana s komponento *String builder*, ki tej doda še drugi del – besedo *holiday*. Na podlagi tega je bila pridobljena nova tako imenovana iskalna povezava, ki jo je bilo treba povezati še s komponento *Fetch Feed*. To nam je omogočilo pridobitev zelenega zadetka (videoposnetka).



Slika 6.5: Pridobivanje videoposnetkov

Združevanje

Zgoraj opisani trije deli so nato združeni s pomočjo komponente *Union* ter pripeljeni na izhod iz cevi. Ukaz *Run pipes* nam prikaže izdelano storitev.

Testiranje

V namen testiranja je bil v veliko pomoč že vgrajen razhroščevalnik, ki ob izbiri posameznih modulov prikaže proizvedene rezultate. Tako lahko celoten proces razvoja lažje nadzorujemo in zaznamo morebitne napake že v razvojnem okolju. Druga uporabna možnost je zagon izvajanja pipe, kjer poteka interakcija uporabnika s končno storitvijo.

6.2 Prepletena storitev v ogrodju Google Spreadsheets

V tem podpoglavju je opisan razvoj prepletene storitve za izbrano problemsko domeno v ogrodju Google Spreadsheets.

6.2.1 Opis delovanja prepletene storitve

Uporabnik najprej v spustnem seznamu izbere željeno državo. Glede na njegovo izbiro se ta prikaže na zemljevidu, posodobijo pa se tudi informacije o njej – njena koda, površina ter zemljepisna širina in višina. Prav tako se prikažejo aktualne novice o državi ter zanimivosti v povezavi z njo. Te črpamo iz dveh virov, Breaking Travel News [4] in strani BBC Travel [3].

6.2.2 Izdelava prepletene storitve

V tej sekciji bo podrobneje prikazan razvoj prepletene storitve po delih.

Za začetek smo uvozili podatke iz Harvardske podatkovne zbirke [23] (oblike .csv) in jih pretvorili v ustrezen format za prikaz v preglednici.

Pridobivanje osnovnih podatkov

S pomočjo možnosti za validacijo podatkov, smo ustvarili spustni seznam, ki ga sestavljajo imena držav iz zbirke podatkov. Naslednji korak za pridobitev

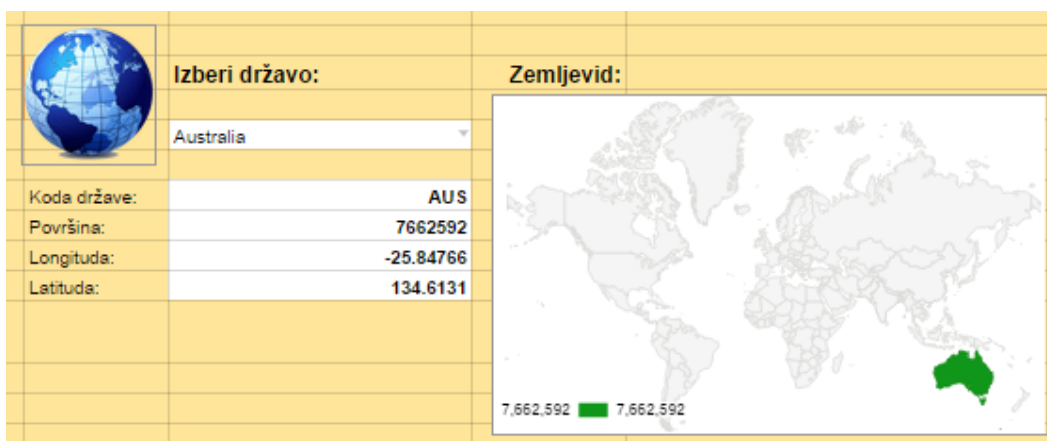
informacij (kode države, površine, zemljepisne širine in višine), je vseboval izvajanje poizvedb s pomočjo funkcije *QUERY* funkcije. Primer ene izmed uporabljenih poizvedb je prikazan spodaj:

```
=QUERY(Podatki!A2:S165,"select A where B=""&D7&""",FALSE)
```

Ta iz naše zbirke podatkov izbere stolpec *A*, ki vsebuje tiste kode posameznih držav, ki ustrezajo vsebini spustnega seznama (ta se nahaja v celici *D7*). Podobne izvedbe se izvedejo tudi za prikaz površine ter zemljepisne višine in širine tako, da spreminjamo vrednost **select** stavka.

Zemljevid je narejen s pomočjo vgrajene funkcije za vstavljanje grafov ter je povezan z imenom izbrane države in njeno površino. Na podlagi prve tudi prikaže lokacijo, ob prehodu miškega kazalca pa pokaže še informacijo o njeni površini.

Obe vrsti omenjenih podatkov se dinamično osvežujeta glede na izbrano državo iz spustnega seznama. Na sliki 6.6 je prikazan primer Avstralije.



Slika 6.6: Prikaz osnovnih informacij o državi in njen prikaz na zemljevidu

Pridobivanje novic

Novice smo pridobivali iz dveh RSS virov s pomočjo funkcije *IMPORTFEED*. Za vsakega izmed njih smo ustvarili ločen zavihek, v katerem smo pridobili dvesto člankov iz specificiranega vira. Primer uporabljene poizvedbe za pridobivanje novic je:

```
=IMPORTFEED("http://www.bbc.com/travel/feed.rss","items",FALSE,200)
```

Aktualne novice		
Datum in ura objave	Naslov	Preberi več
Wed, 10 Sep 2014 08:47:13	News: Malaysia Airlines bookings show signs of recovery	http://feedproxy.google.com/~r/breakingt
Wed, 10 Sep 2014 00:00:31	News: Tourism Australia partners with the Telegraph	http://feedproxy.google.com/~r/breakingt
Sun, 07 Sep 2014 08:00:22	News: Starwood plans W Hotel in Brisbane	http://feedproxy.google.com/~r/breakingt
Zanimivosti		
Datum in ura objave	Naslov	Preberi več
Thu, 14 Aug 2014 18:52:22	The newest spot to swim with whales	http://www.bbc.com/travel/feature/20140
Thu, 01 May 2014 22:17:24	Our favourite images of the week	http://www.bbc.com/travel/slideshow/201
Mon, 21 Apr 2014 19:38:34	A bird that sounds like a chainsaw	http://www.bbc.com/travel/feature/20140

Slika 6.7: Prikaz novic o izbrani državi

Te rezultate smo uporabili tudi za naslednjo poizvedbo, s pomočjo katere smo poiskali tri novice, kjer je v opisu vsebovana država izbrana iz spustnega seznama. Primer poizvedbe za pridobivanje naslova novice je:

```
=query(Novice!A1:D200,"select A where D contains ""&D7&"" limit 3",FALSE)
```

Slika 6.7 prikazuje, kako so pridobljene novice predstavljene z datumom in uro, naslovom ter povezavo do članka.

Informacije o videoposnetkih

Tako kot v ogrođu Yahoo pipes, smo tudi tu želeli pridobiti podatke iz storitve Youtube. Ker smo imeli tukaj na voljo le funkcijo za pridobivanje podatkov

XML oblike *IMPORTXML*, ne pa tudi komponente za gradnjo nizov, smo se odločili videoposnetke o posamezni državi pridobiti s kanala Travel Channel na YouTubeu. Za to smo poleg povezave iskalnega niza morali pridobiti še *XPath* zelenih elementov. Primer poizvedbe *XPath* za pridobitev naslova videoposnetkov:

```
//*[@id="section-list"]/li/ol/li/div/div[2]/h3/a
```

Za njeno pridobitev smo uporabili kar možnost brskalnika Chrome za pregledovanje elementa, kjer imamo nato možnost pridobitve ustrezne poizvedbe *XPath*. Del splošnih rezultatov prikazuje slika 6.8.

Temu koraku je sledil podoben postopek kot pri pridobivanju novic, torej ustrezna poizvedba, ki vrne rezultate z YouTubea za izbrano državo.

<code>//*[@id="section-list"]/li/ol/li/div/div[2]/h3/a</code>	<code>//*[@id="section-list"]/li/ol/li/div/div[2]/div[2]</code>
Naslov	Opis
Travel Channel: Best Destinations in the World	Find out what the best destinations in the World are as seen through the eyes of real travelers. 00:00 - Amsterdam, The Netherlands ...
Travel Channel Wild Carpathia I HD Episodul 1 : TRANSYLVANIA	Charlie Ottley ventures deep into the breath-taking but lit
	mountains and forests of Carpathia. Home to bears, wolv

Slika 6.8: Pridobivanje podatkov o videoposnetkih

6.3 Primerjava implementacije

V tem podpoglavju so opisane ugotovitve ob primerjavi razvoja prepletene storitve v ogrodjih Yahoo Pipes in Google Spreadsheets. Prvi del opisuje splošne ugotovitve in glavne razlike pri realizaciji posameznih delov storitve, medtem ko je v drugem delu predstavljeno, kako se je ocena teh dveh ogrodij po posameznih kriterijih odločitvenega drevesa izkazala še v praksi.

Z obema ogrodjema smo uspešno dosegli realizacijo zastavljene problemske domene. Glede končnega izgleda prepletene storitve smo bolj omejeni z ogrodjem Yahoo Pipes, kjer lahko vplivamo le na vključitev posameznih elementov in vsebine, ne pa tudi na njihovo pozicijo, velikost in oblikovanje vmesnika.

Pri Google Spreadsheets je teh možnosti več, zaradi česar imamo večji nadzor nad samim izgledom.

Implementacija zemljevida se je izkazala za enostavnejšo v ogrodju Google Spreadsheets. Slednje je namreč iz naših podatkov na podlagi imena države to uspešno prikazalo na zemljevidu, medtem ko sta v Yahoo Pipes v ta namen uporabljeni zemljepisna višina in širina. Slabost Google Spreadsheets pri tej realizaciji pa je, da je za prikaz dodatnih informacij v oblaku lahko prikazana le ena sama vrednost.

Yahoo Pipes je omogočal lažje vključevanje novic iz različnih virov, saj je ustrezna komponenta samostojno izluščila najnovejše članke skupaj s sliko ter obenem pridobila povezavo. Google Spreadsheets se je izkazal za bolj omejenega, saj omogoča pridobitev samo besedilnih elementov.

Za pridobitev posnetkov z YouTube smo boljše rezultate dosegli z Yahoo Pipes. Tukaj je bila namreč možna točna določitev iskalne poizvedbe, kjer smo kot rezultat dobili videoposnetek s prikazno sliko, naslovom, povezavo in drugimi informacijami. Pri ogrodju Google Spreadsheets smo lahko posamezne informacije črpali le z uporabo XPath, brez dinamičnega generiranja povezav, kar se je izkazalo za manj učinkovit in zamudnejši postopek.

Kriteriji iz večkriterijskega odločitvenega modela so se izkazali kot dobro zastavljeni, kajti skozi realizacijo storitve, smo lahko tudi praktično ocenili večino. Glede na prvi kriterij uporabniške izkušnje se je bolje odrezal Google Spreadsheets, saj ni bilo pri samem izvajanju storitve nobenih težav. Tudi za začetniško uporabo se je na koncu izkazal za malce enostavnejšega v primerjavi z ogrodjem Yahoo Pipes. Glede podpore uporabnikom se je kljub enako dobri oceni kriterija iz prejšnjega poglavja ponovno bolje izkazalo ogrodje Google Spreadsheets. Pri Yahoo Pipes smo namreč naleteli na nekatere zastarele in nedelujoče praktične primere iz uradne dokumentacije. Pomembno vlogo pri tem kriteriju je igrala tudi možnost diskusije, kjer je v obeh primerih šlo za uporabniški forum. Ta se je tako pri enem kot pri drugem izkazal za uporabnega in je dobro pripomogel k reševanju težav. Kar se tiče funkcionalnosti, je bila

pri obeh ogrodjih razpoložljivost ogrodij nekoliko slabša. Google Spreadsheets z novo verzijo ponuja malce bolj omejeno množico pripomočkov in dodatkov, medtem ko se je pri Yahoo Pipes število razpoložljivih komponent nekoliko zmanjšalo, saj nekatere ne delujejo več (npr. komponenta za Flickr). Kot zelo uporabna funkcionalnost pri ogrodju Yahoo Pipes pa se je izkazala možnost iskanja po bazi. Slednja nam je omogočila zbirko že obstoječih prepletenih storitev, na katerih smo lahko takoj spoznali delovanje različnih kombinacij komponent. Integracija podatkov je v obeh primerih dobro podprta in enostavna, saj nam uvoz podatkov tipa `.csv` v nobenem ogrodju ni predstavljal težav. Za vključitev delno strukturiranih virov se je bolje izkazalo ogrodje Yahoo Pipes, kot je že razloženo v predprejšnjem odstavku.

Poglavje 7

Sklepne ugotovitve

V tej diplomski nalogi je podrobneje raziskano področje ogrodij za ustvarjanje prepletenih storitev. To je narejeno s pomočjo kategorizacije prepletenih storitev glede na njihov tehnološki nivo, okolje izvajanja, namen, ciljne uporabnike ter podporo uporabnikom. Znotraj vsake podkategorije je identificiranih nekaj tipičnih predstavnikov, od katerih je podrobneje opisanih in raziskanih pet. Ti so z večkriterijskim odločitvenim modelom nato ocenjeni na osnovi uporabniške izkušnje, podpore uporabnikom in funkcionalnosti. Njihov rezultat so radarski diagrami, ki ponazarjajo primerjavo ocen vseh kriterijev ogrodij. Za konec je kot potrditev dobljenih rezultatov tudi razvita prepletena storitev z najboljšo ocenjenima ogrođjema.

Na nekaj težav smo naleteli na začetku, ob iskanju ogrodij za ustvarjanje prepletenih storitev. Veliko ogrodij, ki so v strokovni literaturi omenjena kot pomembni predstavniki, je namreč nehalo obstajati. Takšni primeri so Microsoftov Popfly, Intel Mashmaker, IBM QEDWiki ali Google Mashup Editor. Kljub tej oviri pa smo našli dovolj reprezentativne predstavnike, ki so tudi podrobneje raziskani v tej diplomski nalogi.

Diplomsko delo bi lahko nadgradili z razvojem iste problemske domene v vseh izbranih ogrođjih in primerjavo teh procesov. Zanimivo bi bilo tudi primerjati razvoj prepletene aplikacije z lastno sprogramirano problemsko do-

meno ter na ta način ugotoviti, ali lahko z obema postopkoma dosežemo enak rezultat oziroma kakšne so glavne razlike med njima.

Literatura

- [1] *DEXi: A Program for Multi-Attribute Decision Making*, 2013. Dostopno na: <http://www-ai.ijs.si/MarkoBohanec/dexi.html>.
- [2] *Apatar*, 2014. Dostopno na: <http://www.apatar.com/>.
- [3] *BBC Travel*, 2014. Dostopno na: <http://www.bbc.com/travel/>.
- [4] *Breaking Travel News*, 2014. Dostopno na: <http://www.breakingtravelnews.com/news/>.
- [5] *Dapper*, 2014. Dostopno na: <http://open.dapper.net/>.
- [6] *DERI Pipes*, 2014. Dostopno na: <http://pipes.deri.org/>.
- [7] *Google Spreadsheets*, 2014. Dostopno na: <http://www.google.com/sheets/about/>.
- [8] *MarcoFlow*, 2014. Dostopno na: <https://sites.google.com/site/mashtn/industrial-projects/marcoflow>.
- [9] *Mashup*, 2014. Dostopno na: <http://en.citizendium.org/wiki/mashup>.
- [10] *Microsoft Excel*, 2014. Dostopno na: <http://office.microsoft.com/en-001/excel/>.
- [11] *RSSBus*, 2014. Dostopno na: <https://www.rssbus.com/>.

-
- [12] *We Wired Web*, 2014. Dostopno na: <https://wewiredweb.com/>.
 - [13] *Wikipedia*, 2014. Dostopno na: <http://www.wikipedia.org/>.
 - [14] *WireCloud*, 2014. Dostopno na: <http://conwet.fi.upm.es/wirecloud/>.
 - [15] *Yahoo Pipes*, 2014. Dostopno na: <https://pipes.yahoo.com/pipes/>.
 - [16] *YouTube*, 2014. Dostopno na: <http://www.youtube.com>.
 - [17] *YQL*, 2014. Dostopno na: <https://developer.yahoo.com/yql/>.
 - [18] M. I. Ali, R. Pichler, H. Truong, and S. Dustdar. On integrating data services using data mashups. In *Advances in Databases*, pages 132–135. Springer, 2011.
 - [19] A. Auinger, M. Ebner, D. Nedbal, and A. Holzinger. Mixing content and endless collaboration–mashups: Towards future personal learning environments. In *Universal Access in Human-Computer Interaction. Applications and Services*, pages 14–23. Springer, 2009.
 - [20] M. Dojchinovski. Diplomsko delo: Razvoj sestavljanek z jezikom emml. Univerza v Mariboru, 2010.
 - [21] N. C. Engard and J. Levine. *Library mashups: exploring new ways to deliver library data*. Information Today, Incorporated, 2009.
 - [22] T. Fischer, F. Bakalov, and A. Nauerz. An overview of current approaches to mashup generation. In *Wissensmanagement*, pages 254–259. Citeseer, 2009.
 - [23] Center for International Development at Harvard University. Research datasets, 2007. Dostopno na: http://www.cid.harvard.edu/ciddata/geog/physfact_rev.csv.

-
- [24] J. J. Hanson. Mashups: Strategies for the modern enterprise. Addison-Wesley Professional, 2009.
 - [25] Dat Dac Hoang, Hye-young Paik, and Boualem Benatallah. An analysis of spreadsheet-based services mashup. In *Proceedings of the Twenty-First Australasian Conference on Database Technologies-Volume 104*, pages 141–150. Australian Computer Society, Inc., 2010.
 - [26] V. Hoyer and M. Fischer. Market overview of enterprise mashup tools. In *Service-Oriented Computing-ICSOC 2008*, pages 708–721. Springer, 2008.
 - [27] A. Koschmider, V. Torres, and V. Pelechano. Elucidating the mashup hype: Definition, challenges, methodical guide and tools for mashups. In *Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web at WWW*, pages 1–9, 2009.
 - [28] Xuanzhe Liu, Yi Hui, Wei Sun, and Haiqi Liang. Towards service composition based on mashup. In *Services, 2007 IEEE Congress on*, pages 332–339. IEEE, 2007.
 - [29] G. Di Lorenzo, H. Hacid, H. Paik, and B. Benatallah. Mashups for data integration: An analysis. 2008.
 - [30] E. Ort, S. Brydon, and M. Basler. Mashup styles, part 1: Server-side mashups. *Sun Technical Article*, 2007.
 - [31] E. Ort, S. Brydon, and M. Basler. Mashup styles, part 2: Client-side mashups. *Sun Technical Article*, 2007.
 - [32] M. A. Paredes-Valverde, G. Alor-Hernández, A. Rodr’guez-González, and G. Hernández-Chan. Developing social networks mashups: An overview of rest-based apis. *Procedia Technology*, 3:205–213, 2012.

- [33] Jin Yu, Boualem Benatallah, Fabio Casati, and Florian Daniel. Understanding mashup development. *Internet Computing, IEEE*, 12(5):44–52, 2008.
- [34] N. Zang and M. B. Rosson. What’s in a mashup? and why? studying the perceptions of web-active end users. In *Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008. IEEE Symposium on*, pages 31–38. IEEE, 2008.